HOLOGRAPHIC TRANSMIT REFLECT ARRAY AT 5.8 GHZ FOR WIRELESS POWER TRANSFER IN THE FRESNEL ZONE

A Dissertation
Presented to
The Academic Faculty

by

Emily Mei Backer

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
Electrical and Computer Engineering

Georgia Institute of Technology
May 2019

HOLOGRAPHIC TRANSMIT REFLECT ARRAY AT 5.8 GHZ FOR WIRELESS POWER TRANSFER IN THE FRESNEL ZONE

Approved by:

Dr. Gregory Durgin, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Andrew Peterson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Lukas Graber
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 24, 2019

ACKNOWLEDGEMENTS

I would like to especially thank my mother and father who have supported me throughout my college career. Thank you to my fiancé for always having faith in me. I am very grateful for all the people in Dr. Gregory Durgin's lab especially Mohammad Alhassoun, Cheng Qi, and Mike Varner. My advisor, Dr. Durgin has been a wonderful mentor and has provided great insight. I would also like to thank the William C. Brown Microwave Power Transmission (MPT) fellowship and a grant from the Space Solar Power Institute for giving me the opportunity to continue my education in graduate school.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ABC	Absorbing Boundary Condition
F	Focal distance
FE-BI	Finite Element Boundary Integral
FEM	Finite Element Method
γ	Gamma Ratio of Focal Point to Upper Fresnel Zone Bound
IE	Integral Equation
λ	Wavelength
L	Length of the Aperture
MoM	Method of Moments
PML	Perfectly Matched Layer
r	Distance Away
SSP	Space Solar Power
SWIPT	Simultaneous Wireless Information and Power Transmission
θ	Theta Angle off the X axis
WPT	Wireless Power Transfer

SUMMARY

This thesis presents the design and simulation of four square holographic reflectarrays consisting of circular patches for wireless power transfer (WPT) in the Fresnel region. The design includes the calculation of the phase reflection response from different variations of patches and substrate sizes, the calculation of the phase front needed to form a focal point, the simulation of a reference wave, the design of the hologram, and the simulation of the final holographic system. The first set of arrays use FR4 substrate and are made up of 4096, 2304, and 576 number of patch segments arranged in a rectilinear grid, and the last array uses a Rogers substrate and is made up of 4096 patches. These 1.216-meter FR4, 0.912-meter FR4, 0.456-meter FR4, and 1.216-meter Rogers holographic reflectarrays successfully collimate the incoming power into beams at 10-meter, 5-meter, 3-meter, and 10-meter focal points respectively.

CHAPTER 1. INTRODUCTION

Today, Wireless Power Transfer (WPT) is used for a variety of purposes from wireless cellphone charging to space applications. WPT is used in a broad range of applications, including biomedical [1-3], industrial [4], automotive [5], defense [3], space [6-8], and consumer [2, 9-14]. In addition, there are different ways to execute WPT, each with their own strengths and weaknesses. This research focuses on WPT in the Fresnel zone. The technique demonstrates a way to collimate a beam of power which has the potential for important applications in space and on earth.

1.1 Antenna Field Regions

WPT can be executed in the reactive near-field region, the radiating near-field (Fresnel) region, and the far-field (Fraunhofer) region. For large antennas, where the aperture diameter D is greater than 2.5 wavelengths, the radiating near-field starts at $0.62\sqrt{\frac{D^3}{\lambda}}$ and the far field region begins at $r = \frac{2D^2}{\lambda}$ [15]. A diagram of the different regions is shown in Figure 1 below [16].

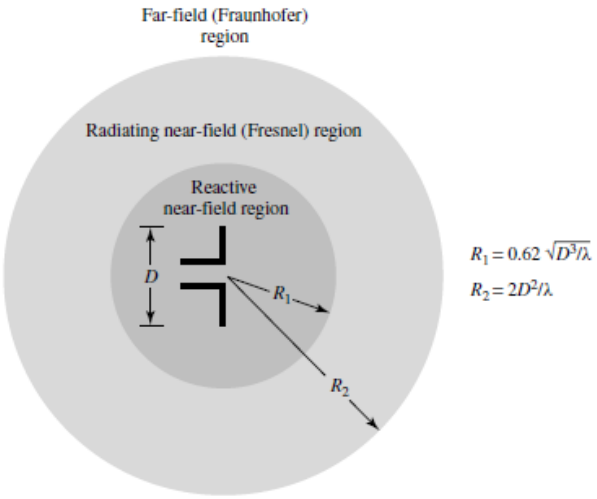


Figure 1 – Antenna field regions

1.1.1 Reactive Near-Field Region

In the reactive near-field region, high efficiencies are achieved and are useful for applications such as portable electronic devices or biomedical implants. However, these applications are very short-range because the available power in the electromagnetic fields falls off at the rate of $\frac{1}{r^6}$. Most experiments achieve distances on the order of, with ranges comparable to the lateral size of the WPT system [17, 18]. Although one experiment [19] uses magnetically coupled resonant structures to achieve a constant 70% efficiency over a larger distance than most inductive coupling techniques, the operating range is still only about 70 centimeters.

1.1.2 Far-Field Region

The far-field offers the ability to transmit at significantly longer distances than the near-field. For example, large amounts of continuous power can be obtained using solar photovoltaic arrays [20]. Space Solar Power (SSP) programs can transmit distances of

36,000 kilometers, with an efficiency of 45% [6]. However, sophisticated equipment must be used for alignment purposes to maintain a line of sight connection, and the receiving aperture must be very large to account for the divergence off $\frac{1}{r^2}$ of the transmitting beam.

1.1.3 Fresnel Region (Radiating Near-Field)

Using the near-radiating field or Fresnel zone is another approach to create WPT systems. The Fresnel zone allows an aperture of size *D* to focus radiation into a specific focal point within the near-field radiation region limit $r = \frac{2D^2}{\lambda}$. The idea is to control the phase of the radiating antenna elements such that they will add in phase at the focal point. Fresnel zone focusing allows an increase by a factor of up to 66.8% in the received power level when compared with the conventional beam-forming antenna [21]. Successful near focused arrays have already been built and tested [21-24].

1.2 Design and Background Theory

The objective of this research is to design and simulate a holographic transmit reflectarray for a WPT demo in the Fresnel zone. For an example demonstration, the reflectarray could reflect and transmit enough energy to power electroluminescence light coils or an LED screen. A WPT demo would comprises a source, transmit antenna, transmit reflectarray, receive antenna, and an electroluminescence light coil. A 5.8 GHz magnetron that outputs 700 Watts could be used as the source with a horn attached as the transmit antenna. The magnetron would illuminate the reflect array off the *X* axis at an incident angle theta θ, with respect to the surface normal of the reflector. The holographic reflectarray then focuses the beam at a focal point *F*, a chosen distance away from the

transmitting aperture to be collected by the receiver. A diagram of the theoretical setup is shown in Figure 2 below. This research will focus on the design and simulation of the transmit reflectarray.

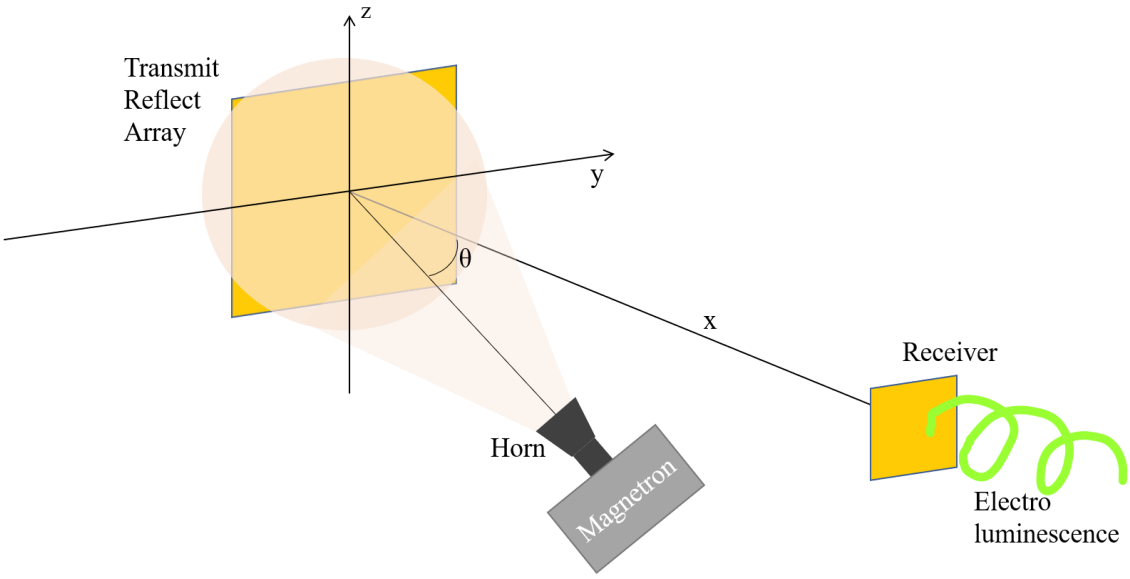


Figure 2 – Theoretical Setup of a Wireless Power Transfer Demonstration

As mentioned previously, the Fresnel zone offers a unique opportunity to transmit power in a focused beam. In the Fresnel region, the aperture behaves similarly to a lens, which can concentrate light into a beam that has a minimum spot or waist size at the focal distance [25, 26]. By using lens geometry, appropriate phases for the array elements can be calculated [23]. To achieve phase convergence at a distance *F* away, Equation 1 below shows that the following phase profile is needed, assuming the aperture is in the *XY*-plane and the focal distance *F* is on the *Z* axis at *Z* = *F* [21, 24, 27].

$$\varphi(x,y) = \frac{2\pi}{\lambda}\left(\sqrt{x^2 + y^2 + F^2} - F\right) \tag{1}$$

Using Equation 1, each element's phase can be determined. Because a magnetron will be used as the power source and is not directly controlling the phase of each element with a feed [21, 23, 24], the transmit array will use holographic metasurface antenna elements to obtain the correct phase at each element. This method avoids the lossy microwave feed networks of a conventional planar array.

The goal is to create a focused beam at a certain distance F away from the aperture. Because the phases at the output of the transmit aperture are known, the desired aperture pattern can be modeled as the interference pattern (IP) between the reference wave and the desired image. This concept comes from Gabor holograms which photographically record the IP between a coherent wave and the fields scattered by an object [28]. If the hologram is illuminated by a copy of the reference wave, the result is a virtual image of the original object. In the experiment, the horn antenna connected to the magnetron will output the reference wave, and the ideal phase distribution necessary to yield a focused beam is shown in Equation 1. If the reference wave is represented by E_R and the desired phase distribution at the output of the aperture is E_A, then the hologram H can be found by [29] shown in Equation 2.

$$\tilde{H} = \tilde{E}_A \cdot \tilde{E}_R^{-1} \tag{2}$$

Metasurfaces provide a unique way to record the IP. Metamaterials are structures that resonantly couple to electric and/or magnetic components of incident electromagnetic fields and are subwavelength metal or dielectric [27]. They can be assembled as elements in two-dimensional arrays to create metasurfaces. Metasurfaces enable spatially varying optical responses such as scattering the amplitude, phase, and polarization, and they can

mold optical wave fronts into shapes [27]. A metasurface can scatter a reference wave with a desired phase when the resonant frequency of each element is carefully designed [29], since resonant metasurface elements can be modeled as resonators due to the internal capacitance and inductance [30].

There are a few challenges associated with the approach of using metasurfaces. One, the amplitude is linked to the phase. By tuning the resonance frequency and therefore the phase, the amplitude and phase change as shown in [25]. This results in a challenge—the amplitude falls to zero at extreme values of plus or minus 90^{o}, which constrains the phase range. However, [29] has shown that by adding a ground plane and operating in the reflective mode, a phase range of plus or minus 150^{o} can be achieved. Another challenge is that the phase varies rapidly near the resonance; thus, an accurate model and fabrication of the elements are necessary. Recent research shows that these challenges can be overcome to enable practical designs [29, 31].

A reflect array comprised of more than 4,000 square metal patches was designed and constructed in [29], and achieved a total efficiency of 25%, which includes the receiver dish efficiency. This thesis goes through the design and simulation and the beginning fabrication of a holographic reflectarray comprising 4096 patches. The array is constructed with circles instead of squares and made from a thinner FR4 substrate and a Rogers substrate.

1.3 Future Applications and Motivation

Since the first concept and design in [32], reflectarrays have improved and now have the potential to improve WPT in many applications. Metasurfaces have allowed arrays to

be very thin and lightweight, and with the addition of active elements, can be electronically steerable reflectors [33]. With the combination of the two principles, holographic reflect arrays can be modified by adding active elements into the array to make them dynamically reconfigurable. One possible way to achieve this is through the use varactor diodes. Varactor diodes change their capacitance and series resistance as the bias applied is varied, and the capacitance change can be utilized to achieve frequency or phase changes [34]. They have been widely used to tune frequency or vary the reflection phase [33, 35, 36].

A dynamically reconfigurable holographic reflect array can be applied in several important space applications including SSP and satellite-to-satellite power and/or communications. For SSP, a theoretical study was conducted by [8] that shows the use of a phase-conjugation array (which behaves like a near field focusing antenna) for beam powering in the Fresnel zone must have an aperture of at least 1 km by 1 km. An existing Fresnel-zone focusing example for use in space and ground systems is a high-gain beam steering system using phase transformation through metasurfaces with a base antenna behind it [37]. Another example with the use of metasurfaces in space applications is a reconfigurable Fabry-Perot cavity antenna that contains a reconfigurable metasurface which can steer its beam to an off-normal direction [7]. A reconfigurable holographic array would provide an excellent beam collimator for use in SSP that could be lightweight, potentially foldable, flat and very thin. These arrays could also be used in power and communications for satellites in space. Reconfigurable holographic arrays could provide power to other satellites in space, and in doing so, other satellites would no longer need solar arrays.

Drones have become a rapidly growing industry in defense and civilian markets. They are used for surveillance, tracking, offensive operations, cinematography, racing, and potentially package delivery. However, drones have a limited operating time and need frequent charging for continued use. Much research has explored the use of platforms that wirelessly charge drones whenever they land. Some examples are in [11-13]. [9, 14] demonstrated that drones can be charged while hovering in the vicinity of a charging pad. These WPT systems require drones to have fixed positions or hover within centimeters of the recharging system. Other research investigated using a drone network that uses a Simultaneous Wireless Information and Power Transmission (SWIPT) scheme through the use of harvesting circuits [10]. If the holographic array is made with active elements and could steer its beam, it could potentially be used to charge drones in mid-air at far distances, similar to [10]. It would be able to keep its power beam pointed on the drone and follow it to constantly charge it. The input power supply and position could stay constant, and the holographic array would collimate the power into a beam pointed at the drone as it flies or hovers.

CHAPTER 2. SIMULATION OF SINGLE PATCH

On the holographic array, small patches of copper above a copper ground plane will be used to construct the metasurfaces. By varying the size of these patch, the phase and reflection coefficient of an incoming wave can be altered on reflection. To examine the reflection response from varying the patch size, a single patch was simulated in ANSYS HFSS using Floquet boundary conditions and master and slave boundaries. Floquet boundary conditions allow for a simpler and easier simulation of a model, and the master and slave boundaries are a feature in HFSS that allow for a proper setup of the model. For a large planar-periodic structure, a Floquet port can be used to simplify the simulation. The analysis of a large planar-periodic structure can be idealized as infinitely large, and as a result, only analysis of a single unit cell with boundaries on the side walls and a Floquet boundary on the top to account for the infinite space above is needed [38]. An excited Floquet boundary can be pictured as a plane wave incident on an infinite array from above. The final array will be large with planar-periodic structures. Although the patches wewere not the same size across the array, [29] used Floquet boundary conditions for simulation and had successful results. All the graphs use the deembed distance setting except for some graphs in Figure 11. The deembed settings box is explained in Section 2.2.3.

2.1 Square Patch

Initially, a single square patch with a 3 mm thick FR4 substrate was simulated in HFSS as shown in the Figure 3 below to compare with the results in [29].

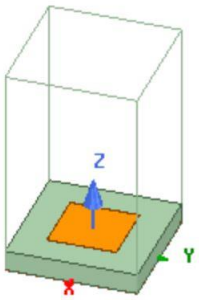


Figure 3 – Setup for Square Patch Simulation

The air box around this basic cell has two master and slave boundaries on the sides and a Floquet port on the top. To achieve off-axis radiation, the theta scan angle between the master and slave boundary was set to 25 degrees. A parametric setup was created to vary the patch size from 1.2 mm to 16.5 mm in steps of 0.1 mm. The step size was chosen as 0.1 mm to make the simulation faster. Table 1 below shows the characteristics of the patch. Note that the patch is in the *XY* plane, and the *Z* coordinate is the patch height as shown in Figure 3.

Table 1 – Square Patch Characteristics

Part	GND_Plane	Substrate	Patch	Airbox
Dimensions (X×Y×Z)	19 mm × 19 mm × 1.4 mil	19 mm × 19 mm × 60 mil	10 mm × 10 mm × 1.4 mil (varied in parametric setup)	19 mm × 19 mm × 27.9117 mm
Characteristics	Material: Copper Shape: box	Material: FR4 Shape: box	Material: Copper Shape: box	Material: Air Shape: box

Figure 4 below shows the reflection response as the patch size is varied from 1.2 mm to 16.5 mm. The reflection response is the magnitude and the phase in degrees of the

FloquetPort1:1, FloquetPort1:1 parameter, which is effectively the S11, or the reflection response.

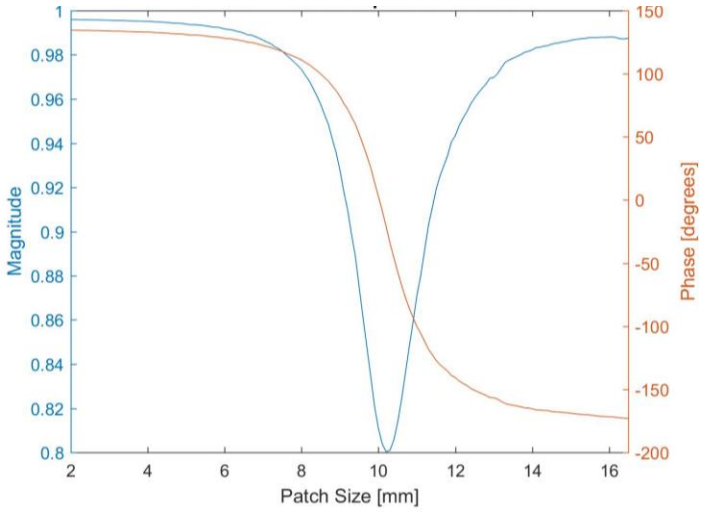


Figure 4 – Reflection Response vs. Square Patch Size with 3 mm Substrate Height and Floquet Port at 26.34 mm Away

The results are very similar to [29], i.e. 307.8 degrees compared with 300 degrees in [29]. However, the range of phase start and end is slightly different, varying from 134.8 degrees to -173.01 degrees compared with 150 degrees to −150 degrees in [29]. The phase varies the most between 6 mm and 14 mm. On a side note, to achieve the smooth graph in Figure 4, in the Solution Setup Settings dialog box, the Minimum Number of Passes and the Minimum Converged passes was set to 5. Otherwise, the graph will have random jumps.

2.2 Circular Patch

One of the important points this research examined was the effect that circular patches has on the polarization of the array. Therefore, the square patch was changed to a circular patch to look at the reflection response of circular patches. In HFSS, the shape of the patch was changed from a box to a cylinder with the radius as the variable that changes.

Changing the patch from a square to a circle shifts the reflection coefficient and phase to the right. Figure 5 below shows a graph of the reflection response vs. patch size for a circular patch with a substrate height of 3 mm and a Floquet port 26.34 mm away with the deembed option checked.

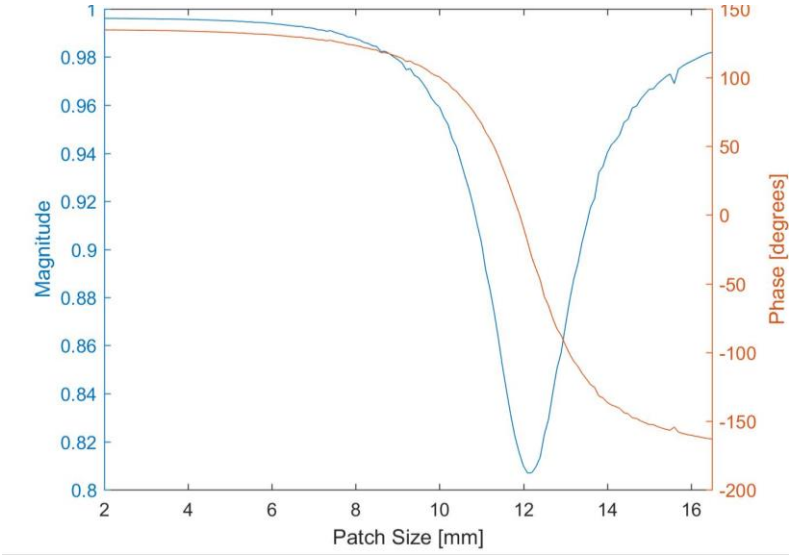
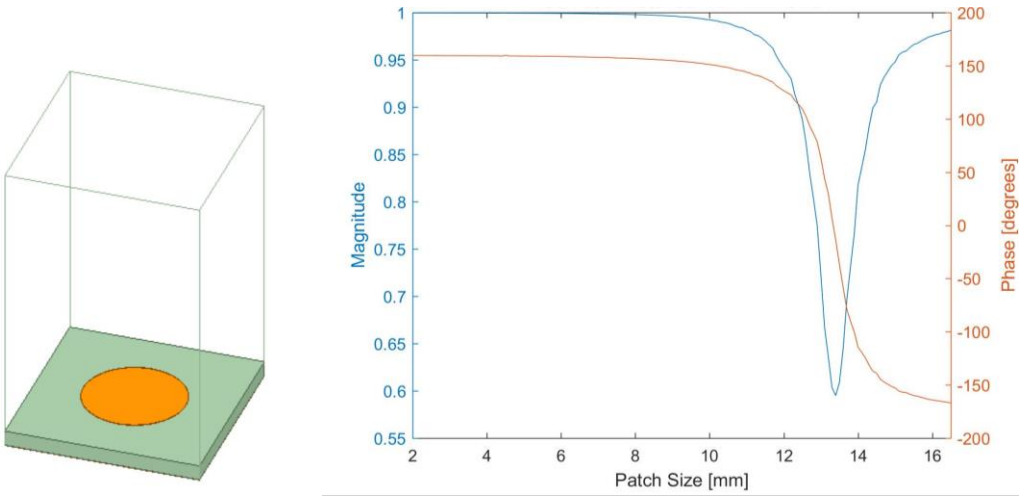


Figure 5 – Reflection Response vs. Circular Patch Size with 3 mm Substrate Height and Floquet Port 26.34 mm Away

The phase now changes most rapidly from 8 mm to 16 mm with a phase range of 134.89 degrees to −168.02 degrees.

Because 60 mil FR4 is less expensive, the substrate height in the simulation was changed. The following Figure 6 shows the circular patch setup and the reflection response vs. patch size for a circular patch with a 60 mil FR4 substrate height.



(a) (b)

Figure 6 – (a) Circular Patch HFSS Setup, (b) Reflection Response vs. Circular Patch Size with 1.524 mm Substrate Height and Floquet Port 26.34 mm Away

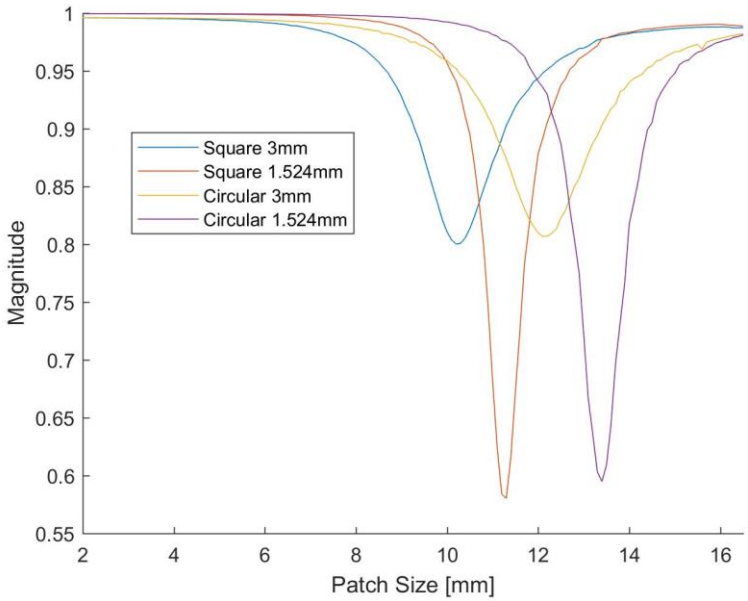
By changing the substrate height, the slope of the phase response is steeper—most of the phase change is now from 11 mm to 16 mm. In addition, the reflection coefficient minimum changed from 0.8 to 0.59 and from 12.2 mm to 13.4 mm. The phase varies from 159.54 degrees to −166.88 degrees. This variation of 326.42 degrees is slightly larger than all the previous phase variations, and is larger if the upper diameter of the circular patch is extended out to 18.9 mm.

Table 2 shows a comparison of all the reflection response graphs from square patch to circular patch to changed substrate height. It also includes a square patch simulation with a 1.524 mm substrate height. The red text represents what was changed from one simulation to the next. For example, from simulation A to B the substrate height was changed.

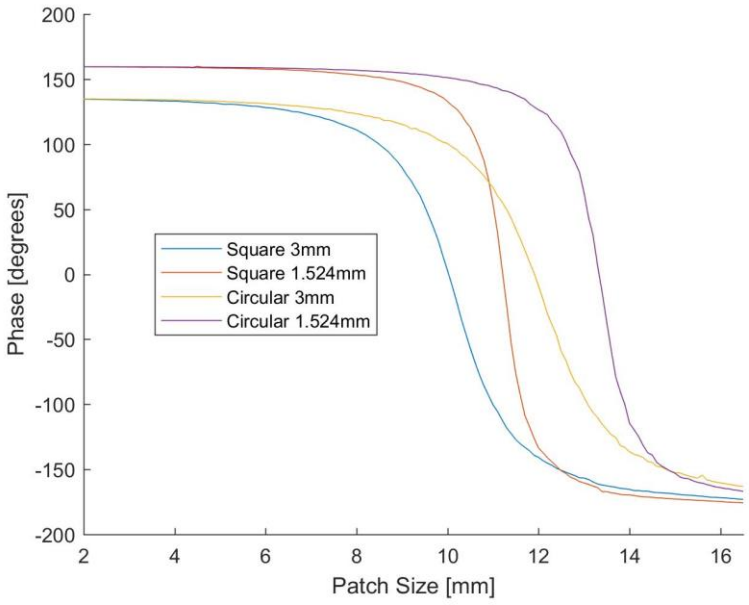
Table 2 – Comparison of Reflection Responses for Varying Patch Sizes with Floquet Port 26.34 mm Away

	A	B	C	D
Patch Type	Square	Square	Circular	Circular
Substrate Height	3 mm	1.524 mm	3 mm	1.524 mm
Phase Range (degrees)	134.79 to -173.01	159.50 to -175.71	134.89 to -168.02	159.53 to -166.88
Reflection Coefficient Minimum	0.800	0.58	0.807	0.595

Figure 7 below shows a graph of the reflection responses of each of the simulations from the above table on the same graph. In Table 2 above, A corresponds to the Square 3 mm SubH blue line, B corresponds to the Square 1.524 mm red line, C corresponds to the Circular 3 mm SubH yellow line, and D corresponds to the Circular 1.524 mm SubH purple line.



(a)



(b)

Figure 7 – Reflection Responses from 3 Simulations in Table 2 (a) - Magnitude of the reflection coefficient, (b) - Phase of the reflection coefficient

Figure 7 (a) shows that changing the shape from a square to a circle shifts the reflection coefficient dip to the right, while changing the substrate thickness shifts the dip to the right and lowers the magnitude by roughly 0.2. Figure 7 (b) shows that changing the shape and substrate thickness shifts the curve to the right. Changing the substrate thickness also makes the curve steeper but gives it a wider range of phase values. As a result, a lower substrate thickness has an advantage of yielding a holographic array that could yield a larger phase range; however, it forces the manufacturing to be very precise because a small error could drastically change the phase response. For instance, for a square patch with a substrate height of 1.524 mm, a change of length from 11.1 mm to 11.2 mm changes the phase from 31.97 degrees to 4.01 degrees which is a 27.96-degree difference. For a circle patch with a substrate height of 1.524 mm, a change in length from 13.3 to 13.4 yields a phase change of 24.19 degrees from 13.87 degrees to -10.32 degrees. The milling

machine must be very precise because phases between 31.97 degrees and 4.01 degrees and phases between 13.87 degrees and −10.32 degrees will most likely be needed. This need for precision may prove to be challenging for manufacturing with a thin FR4 substrate.

2.2.1 Circular Polarization

One of the aspects of this research was to examine the effects of making the patches circular rather than rectangular. Circles generally are more polarization-independent due to their radial symmetry. A short simulation was performed by inserting a far field sphere and examining the axial ratio to get a glance at how the final aperture might behave with respect to circular polarization. Figure 8 below shows the axial ratio as a function of the patch size and theta scan angle (θ), and patch size and phi scan angle (ϕ).

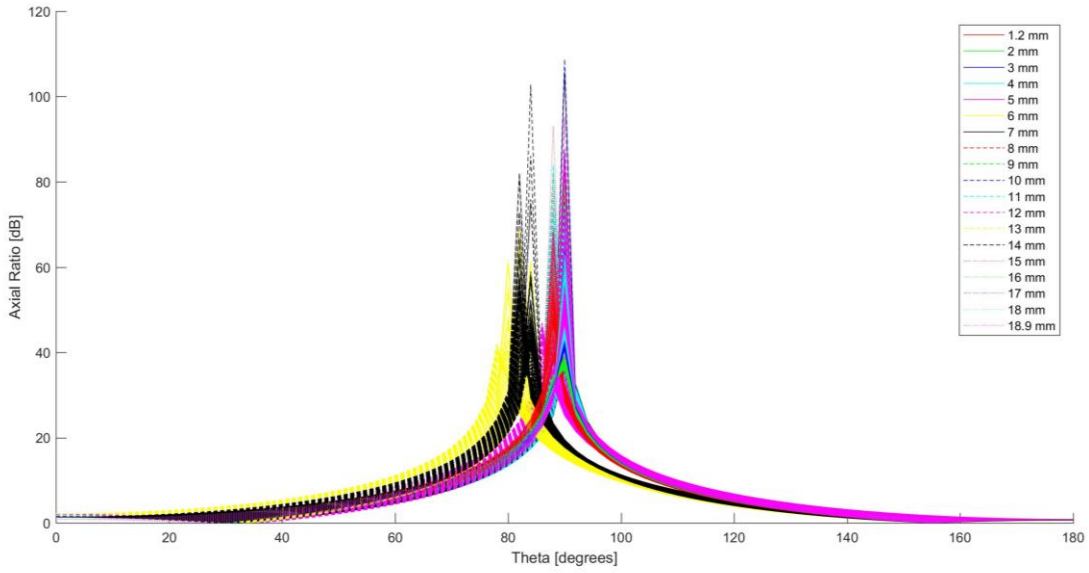


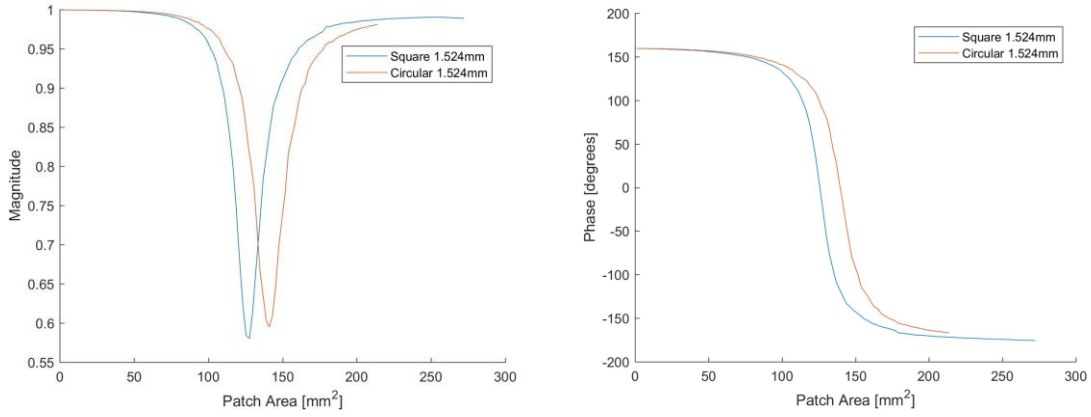
Figure 8 – Axial Ratio

Each color and line style in Figure 8 represent different patch sizes, and all the lines of the same color and line style represent phi angles from −180 degrees to 180 degrees in steps

of 2 degrees. There are 19 different patch sizes that includes: 1.2 mm, 2 mm to 18 mm in steps of 1 mm, and 18.9 mm as shown in the legend in Figure 8. The axial ratio is high only when theta is between 60 and 120 degrees, which makes sense because when theta approaches 90 degrees, the polarization becomes less circular.

2.2.2 Area of Circle and Square Comparison

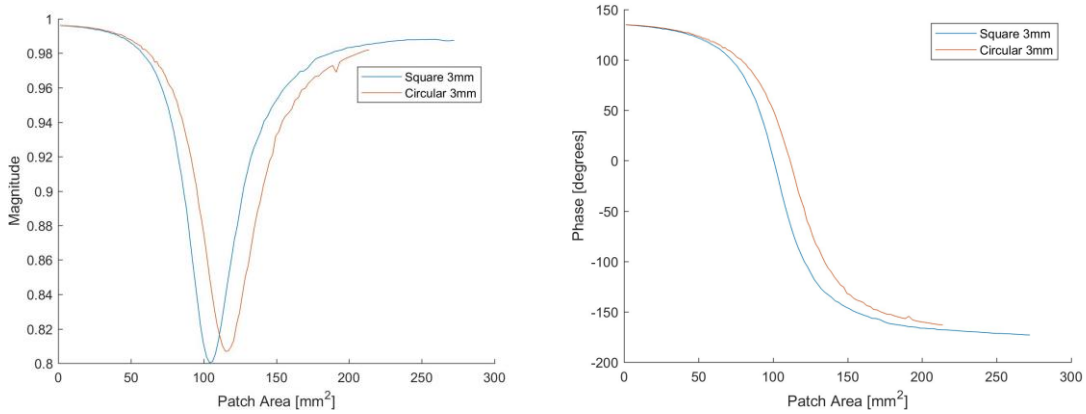
The shift to the right when the shape is changed from a square to a circle is partially be due to changes in the copper area. Figure 9 and Figure 10 below show graphs of the reflection response versus the area of the shape where the square area is L^2 and the circle area is $\pi \left(\frac{L}{2}\right)^2$.



(a) (b)

Figure 9 – Area Comparison of Reflection Responses on 1.524 mm Substrate Height (a) Magnitude (b) Phase

Because the square area is larger for the given L range from 1.2 mm to 16.5 mm, the blue line representing the square patch extends farther than the red line representing the circle patch.



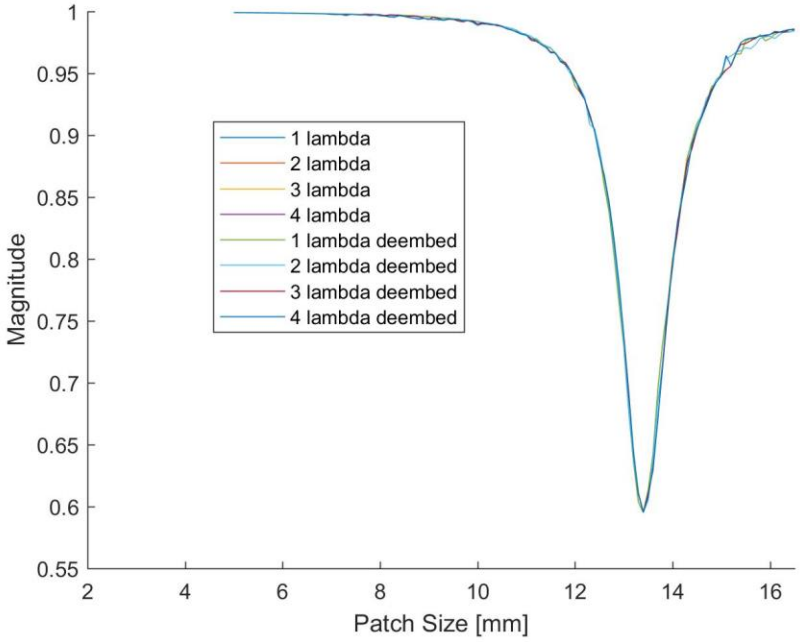
(a) (b)

Figure 10 – Area Comparison of Reflection Responses on 3 mm Substrate Height (a) Magnitude (b) Phase

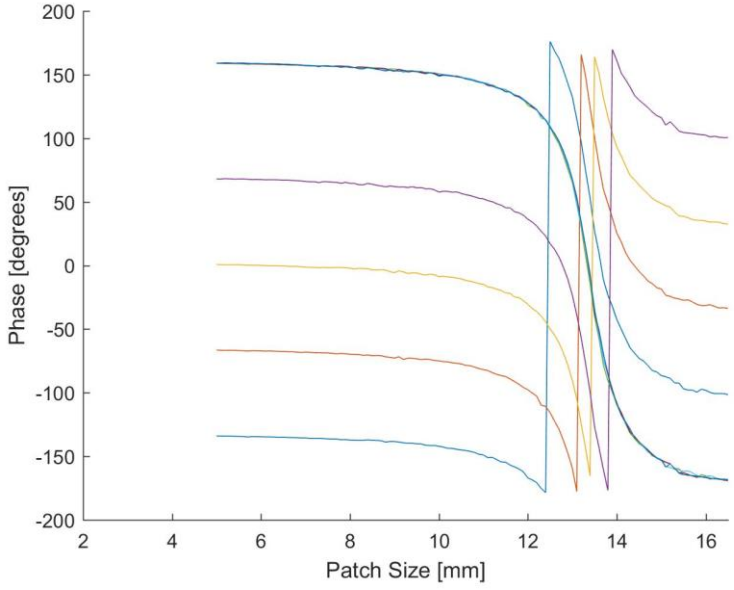
The graph lines are closer together when graphing the reflection response against the area than when graphing the reflection response against patch size in Figure 7; however, they are not exactly overlapping. Thus, both the area and shape of the patch has some effect on where the reflection response curve lies.

2.2.3 Deembed Distance

Changing the distance of the Floquet port from the patch also shifts the graph, but it shifts the phase response curve up and down rather than to the right or left. The following figure shows a series of graphs that demonstrate how changing the distance of the Floquet port shifts the phase curve up and down.



(a)



(b)

Figure 11 – Reflection Response vs Patch Size for Different Distances Away from Patch (a) – Magnitude of reflection coefficient, (b) Phase of reflection coefficient

In Figure 11 (a), the magnitude of the reflection coefficient doesn't change much. However, in (b), because the phase curve changes rapidly, it is important that the deembed button is clicked under the Floquet settings. This deembed button takes out the phase shift

from the distance specified by the user. In this case, the distance from the Floquet port to the top of the patch.

2.2.4 Scan Angle

Changing the scan angle and keeping the patch size constant, changes the phase response. Figure 12 below shows the reflection response for a circular 10 mm diameter patch by varying the scan angle from 0 to 180 degrees in 2-degree steps.

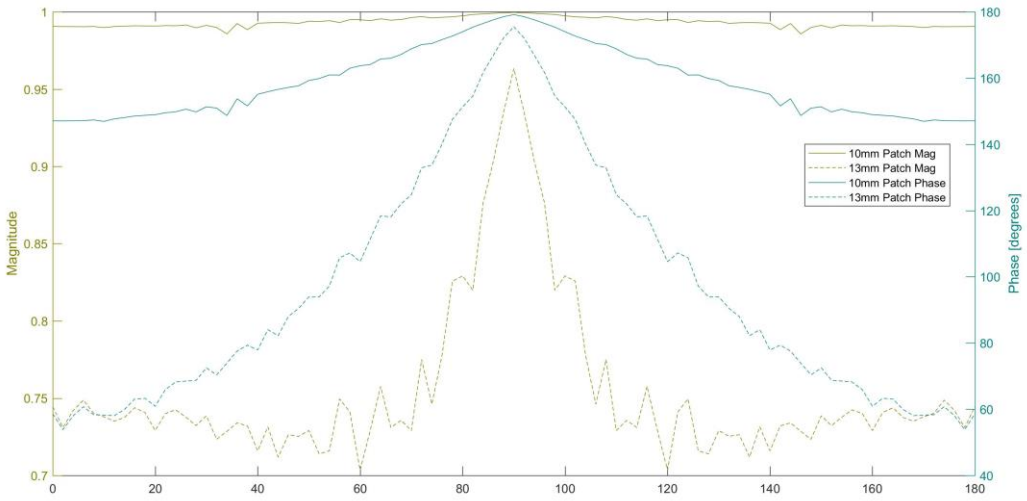
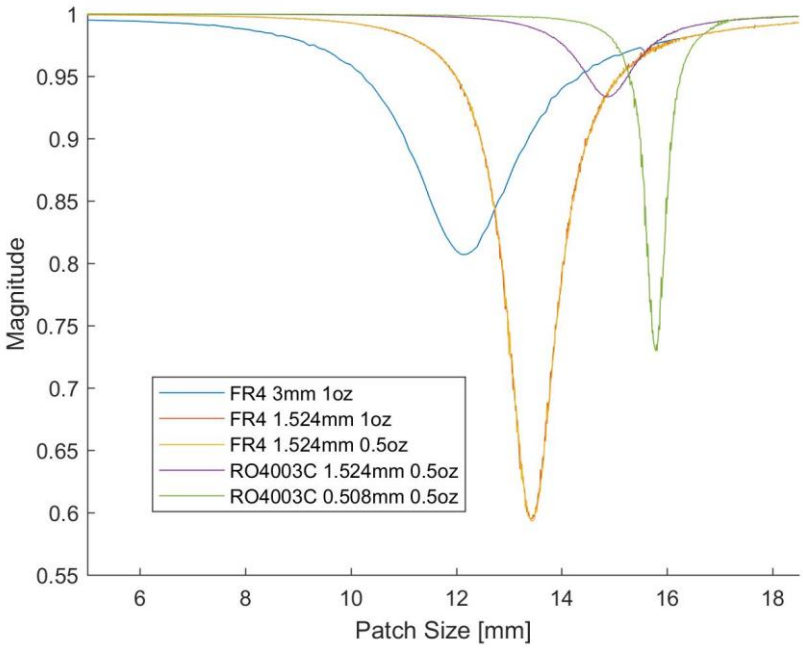


Figure 12 – Reflection Response vs. Theta Scan Angle for 10 mm Diameter Patch with Substrate Height 1.524 mm

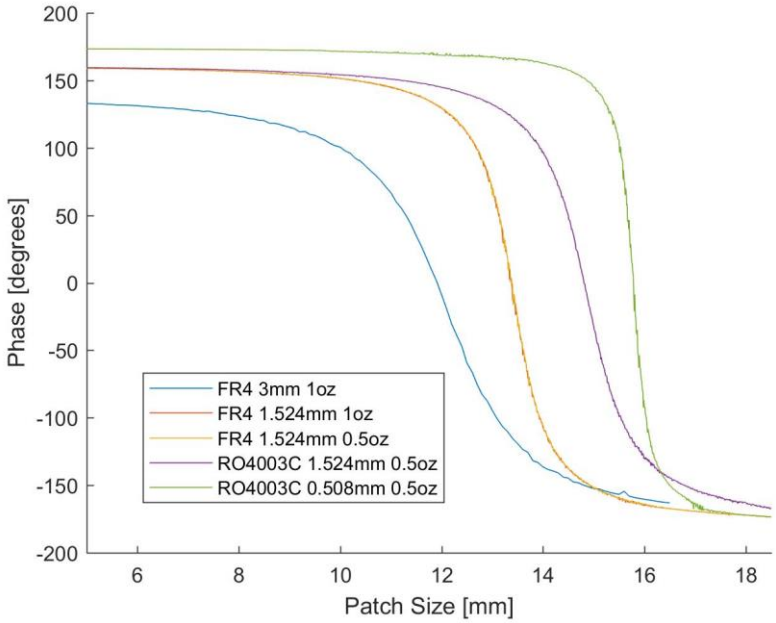
The phase response of the 10 mm patch varies in magnitude from 0.9996 to 0.9857 and in phase from 146.92 degrees to 179.15 degrees. The phase response for the 13 mm patch varies in magnitude from 0.7034 to 0.9632 and in phase from 53.82 degrees to 175.51 degrees. The curve also is symmetrical about a vertical line at 90 degrees. Figure 12 shows that changing the angle of incidence will change the phase and magnitude, but the size of the patch will also affect the extent of the change in the magnitude and phase range. For this holographic array, it is important that the incident angle stay constant.

2.2.5 Rogers Substrate

Rogers substrate, specifically RO4003C, was used to examine the effects of the two different substrate materials. RO4003C is a hydrocarbon ceramic laminate. The following Figure 13 shows how Rogers RO4003C at two different substrate heights affected the reflection curve. Two different thicknesses of copper were used, 1 oz/ft^2 and 0.5 oz/ft^2 which corresponds to about 35 µm or 1.4 mils and 17.5 µm or 0.7 mils respectively.



(a)



(b)

Figure 13 – FR4 vs. Rogers Substrates

The Rogers substrate at 60 mils (1.524 mm) gives a reflection response that is slightly less steep than FR4 with a 60 mils substrate height. In a shallower curve, the phase difference between each patch length is less. This could prove useful if the manufacturing resolution is not small enough to fabricate 0.01-0.02 mm differences in patch sizes. The biggest change in 0.1 mm steps is about 17 degrees as opposed to 24 degrees in FR4. In addition, the magnitude of the reflection coefficient does not dip as much as the FR4, only dipping to 0.93347. This could also make the holographic array slightly more successful. However, Rogers substrate boards can be an expensive.

2.2.6 Final Reflection Response Simulation

Finally, a simulation was completed using a smaller patch step size to obtain a reflection curve that will be used for the rest of the holographic array design. The Floquet port is located 2 wavelengths away from the patch, and the patch size ranges from 1.2 mm

to 18.99 mm in steps of 0.01 mm. Figure 14 below shows the resulting reflection response using a 1.524 mm FR4 substrate.

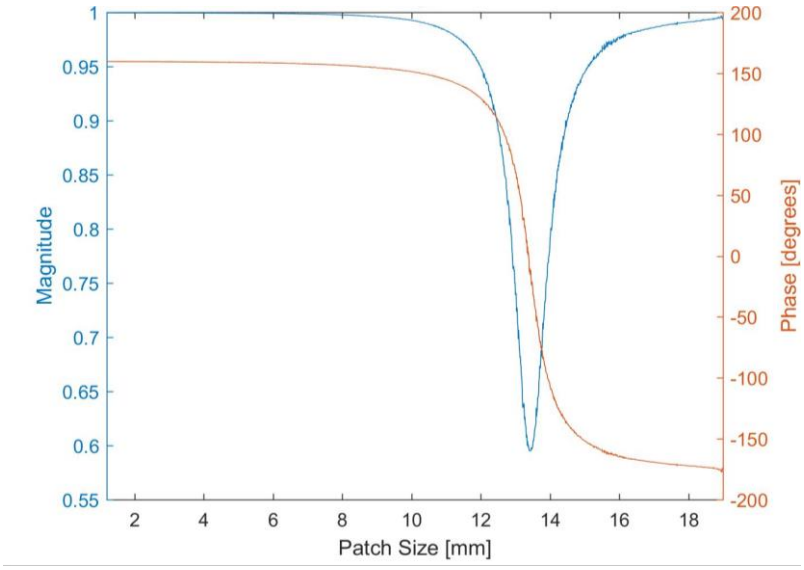


Figure 14 – Reflection Response vs Patch Size Used for Final Holographic Array

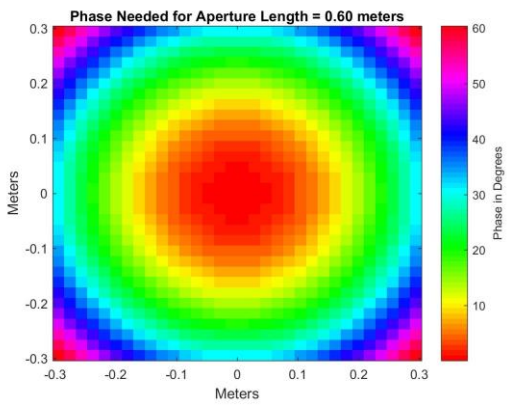
The simulation took about 24 hours; nevertheless, it is important for accurately calculating the necessary patch size for the final hologram reflect array instead of approximating a best fit line. A similar simulation was performed using Rogers substrate to make a Rogers hologram for comparison later.

CHAPTER 3. CALCULATING THE HOLOGRAM

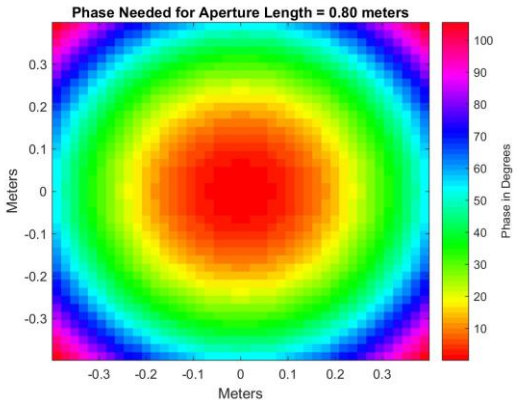
3.1 Phase Calculation

To achieve phase convergence at a distance F away, Equation 1 from section 1.2 shows the phase profile needed [21, 24, 27]. Using this equation, a graph of the phase needed to make a beam that converges at a distance F away can be shown. There are a few parameters that can be changed. One parameter is the aperture size, which will change the phase distribution needed at the output of the aperture. A short MATLAB script was made with Equation 1 and used to make the following Figure 15, Figure 16, and Figure 17.

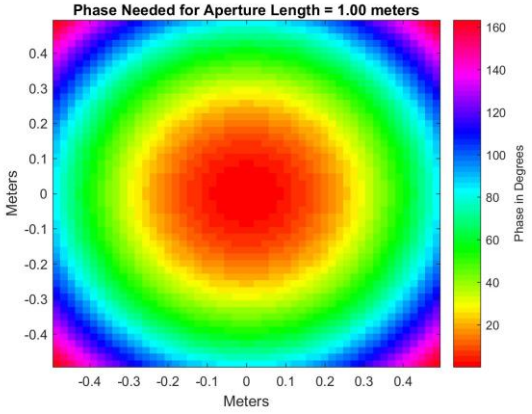
Figure 15 below shows how the phase range will change with a changing aperture size. It shows that for smaller apertures, a smaller phase range is needed. For a 0.6-meter by 0.6-meter aperture, only phases from 0 to 60 degrees are needed, as opposed to a 1.4-meter by 1.4-meter aperture where all phases from −180 to 180 degrees are shown. However, with a smaller phase range the final solution hologram phase needed will still be −180 to 180 degrees.



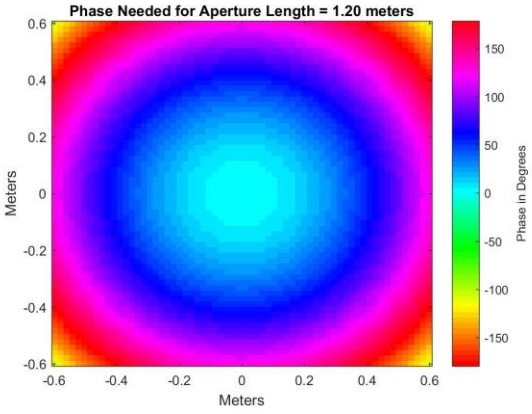
(a) 1.2674 m – 13.9296 m



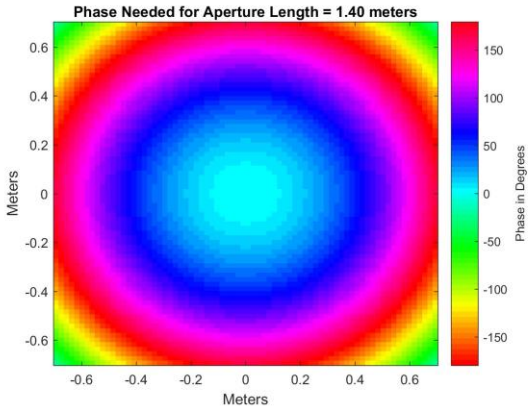
(b) 1.9513 m – 24.7638 m



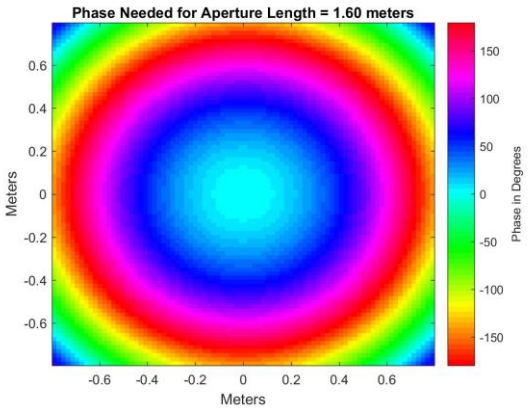
(c) 2.7271 m – 38.6934 m



(d) 3.5848 m – 55.7185 m



(e) 4.5174 m – 75.8391 m



(f) 5.5192 m – 99.0552 m

Figure 15 – Phase Needed for Various Aperture Lengths at Focal Distance F = 10 m. (a) – (f) the aperture length varies from 0.6 m to 1.6 m in steps of 0.2 m. The Fresnel Region is printed below each figure.

In addition to varying the aperture length, an aperture focal distance must be chosen. The following Figure 16 demonstrates how changing the focal distance affects the

phase distribution. The graphs range from 4 meters to 56 meters which is roughly the span of the Fresnel Region for an aperture diameter of 1.216 meters.

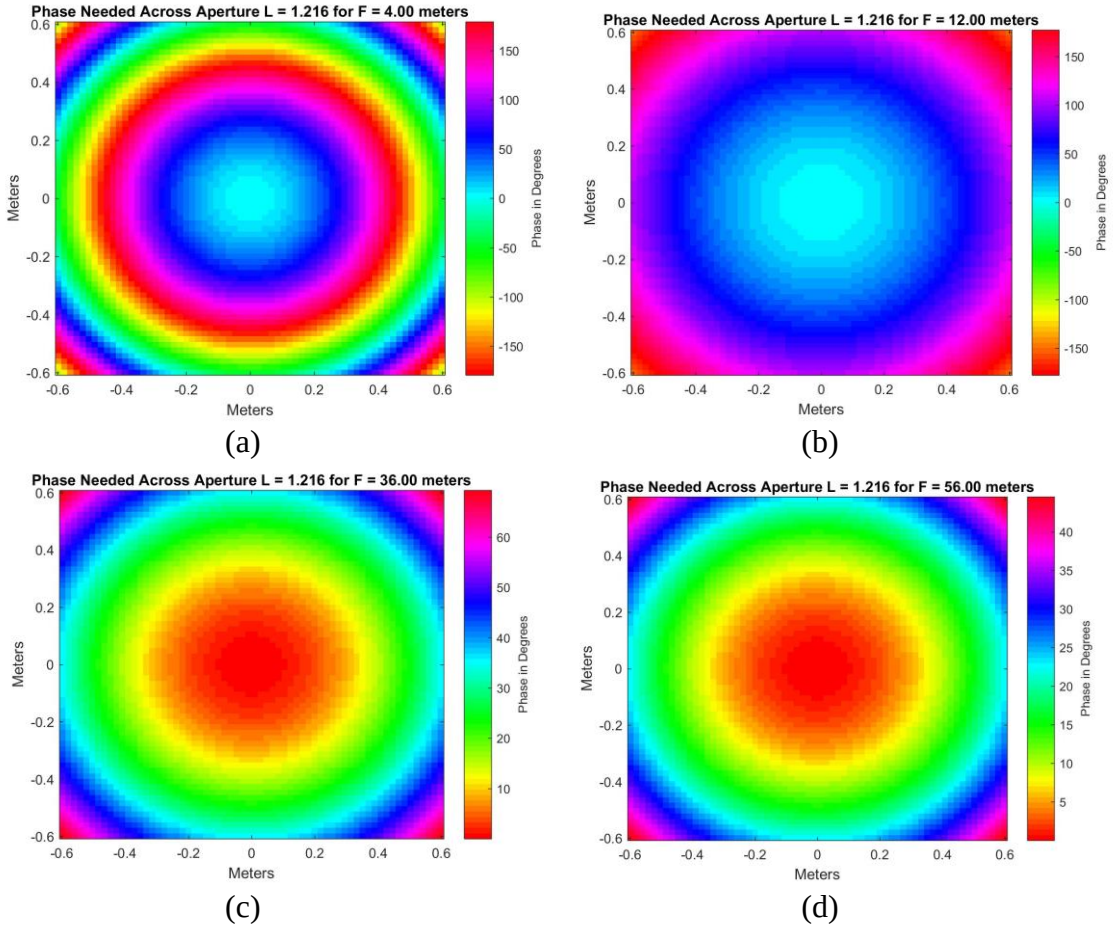


Figure 16 – Phase Distribution for Aperture Length 1.216 m for various Focal Distances. (a) F = 4 m, (b) F = 12 m, (c) F = 36 m, (d) F = 56 m

Figure 16 shows that for a given aperture length, the phase range will decrease with an increasing focal point. Although Figure 16 (c) and (d) may look the same, the phase scale changes from 0 to 60 degrees to 0 to 45 degrees.

When choosing an aperture size and focal point, there are a few important things to consider. Having a small range of phase is not necessarily beneficial because the final hologram array phases will still range between −180 degrees to 180 degrees; In addition,

smaller arrays have a shorter range as seen in the sub captions under each graph in Figure 15. In addition, for a given focal distance, the focal point will approach the end of the Fresnel region as the aperture length decreases. Better near-field focusing is achieved with larger antenna sizes, where the ratio of the focal point to the Fresnel region boundary is small [22]. As a result, once an aperture size has been determined, it is best to keep the ratio of the focal point to the Fresnel region boundary small [22].

An aperture size of 1.216 meters by 1.216 meters was chosen to allow a grid of 64 by 64 patches. With an aperture length of 1.216 meters, the Fresnel region will start at $0.62\sqrt{\frac{D^3}{\lambda}}$ which is equal to 3.6568 meters and end at $r = \frac{2D^2}{\lambda}$, which is 57.2143 meters [16]. The focal distance F can be chosen in between this region; F = 10 meters in the following figure. With a focal distance of 10 meters, the ratio of the focal distance to the boundary of the Fresnel region is 0.1748, which is small. Each pixel in the figure below is equivalent to a 1.9 cm by 1.9 cm square, which is the patch periodicity in the array. The phase ranges from 180 degrees to 0 degrees and -110 degrees to -180 degrees.

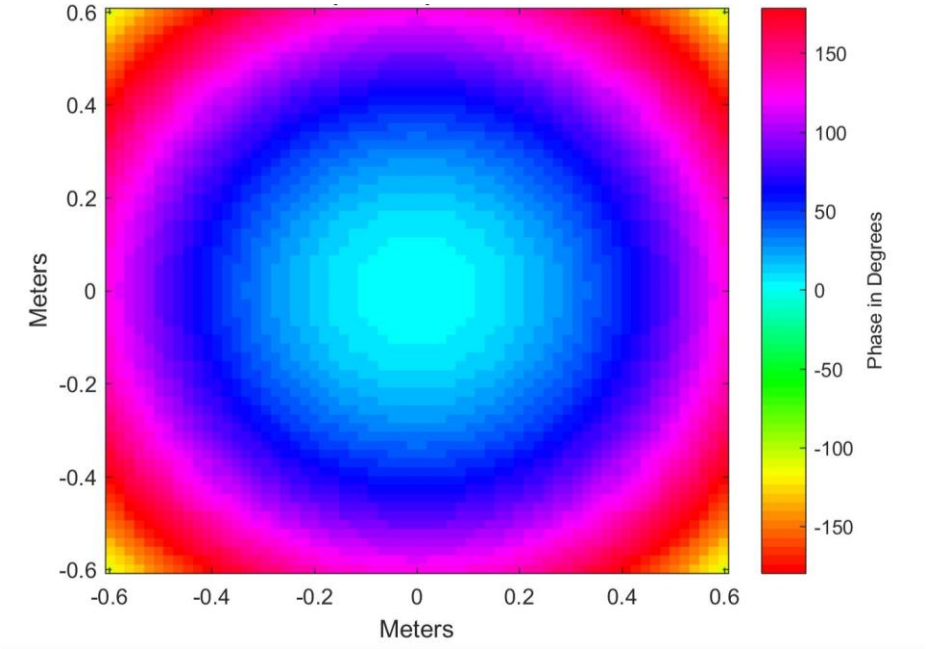


Figure 17 – Phase Needed for an Aperture Length 1.216 m

3.2 Horn Reference Wave

To find out what the wave front looks like at the aperture, a simulation of the reference wave needs to be completed. First, the horn was designed to be as close to the one in the live experiment as possible, and then it was simulated for verification. Once the horn is made, the simulation to determine the incoming horn reference wave at the input of the aperture can be made.

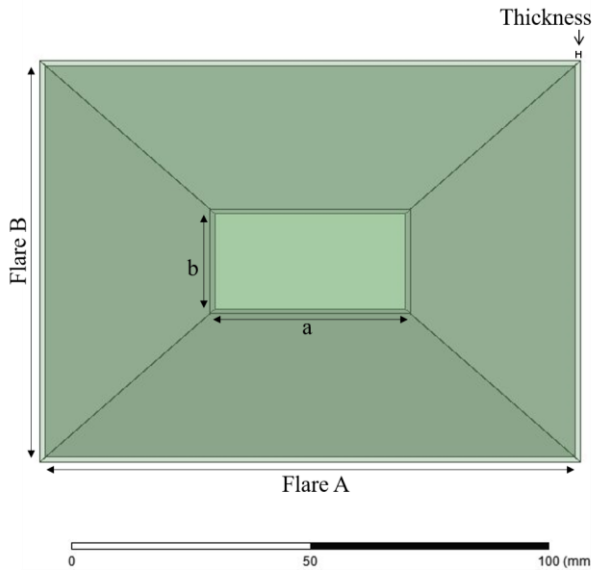
3.2.1 The Horn Antenna

The horn was first modelled in HFSS based on the horn that is in the laboratory attached to the magnetron. The horn model number is PE9860-15, and its specifications can be seen in the datasheet [39]. The dimensions used for the horn are shown in the Table 3 below.

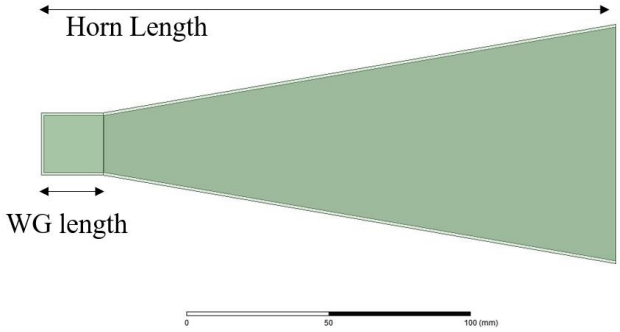
Table 3 – Horn Dimensions

Horn Part	Horn Dimensions (mm)
Horn Length	203.84
Flare A	112.52
Flare B	83.06
a	40.39
b	20.19
WG length	21.35
Thickness	1

The horn was modelled with aluminium in HFSS, but it really has an anodized aluminium body according to the datasheet [39]. Figure 18 below shows the horn with labelled parts from the above Table 3.



(a)



(b)

Figure 18 – Horn Diagram

Figure 19 below shows the simulation results of the horn. The maximum is 15.5 dB which is close to the 14.5 dBi in the datasheet [39].

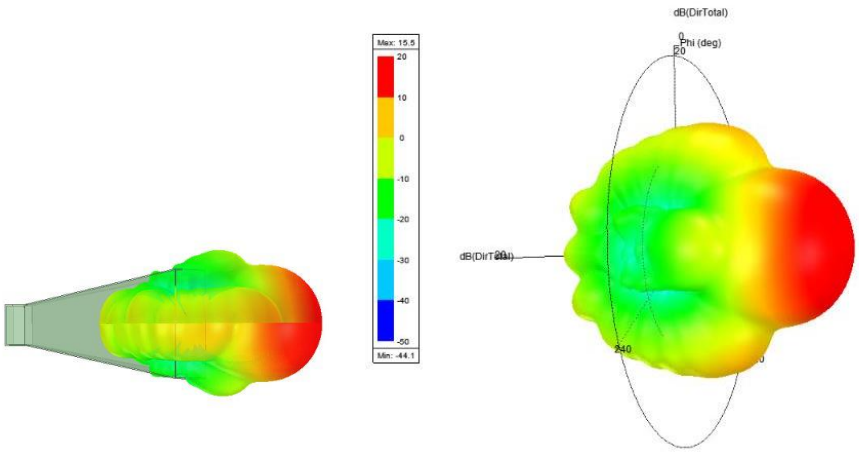


Figure 19 – Horn Simulation Results

3.2.2 Horn Setup Method 1

With the horn working properly, the horn's reference wave was simulated. To make it easier, the horn was exported as a 3D component and inserted into another design in another coordinate system to achieve the off-axis illumination. Note that this simulation will take a large quantity of RAM. Using 32 GB of RAM, the largest simulation that HFSS

could handle approximately was with an aperture of 38 cm by 38 cm with the horn 30 cm away which is about 235837 tetrahedra in a matrix size of 1511533.

The set-up of the horn with the box size that shows the horn 190 cm away from the aperture is shown in Figure 20 below.

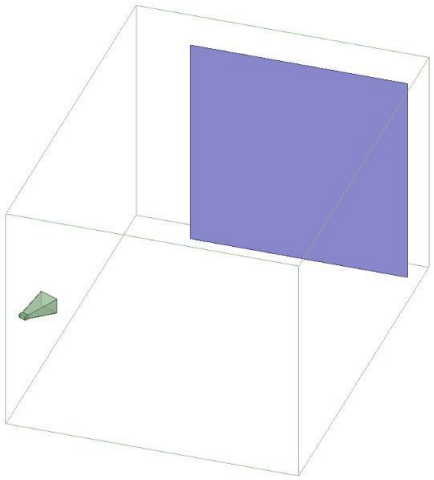


Figure 20 – Horn Reference Wave Simulation Setup 1

The purple aperture is there for visual representation of the plane across which the E field vector should be extracted to plot in MATLAB. During the simulation it should be removed. This simulation is very computationally heavy and requires more than 128 GB of RAM. The matrix size is 67.4 GB of RAM with 6071633 tetrahedra. This simulation is also the "brute force" way to simulate the reference wave pattern. If one has the computational resources to simulate it, it is an easy way to obtain the reference wave pattern of the horn across a plane region within the airbox. In addition, the horn's electric field pattern across a different region of space inside the box would be easily obtainable without re-running the simulation.

3.2.3 FE-BI

HFSS's Finite Element Boundary Integral (FE-BI) is a way to simulate the horn's reference wave that doesn't require as much computational resources. In HFSS, boundary conditions can be used to emulate the free space environment. There are a few ways to truncate infinite free space into a finite computational domain including Absorbing Boundary Condition (ABC), Perfectly Matched Layer (PML), and FE-BI [40]. The boundary conditions make simulations computationally less heavy than using a giant air box radiating boundary by making the solution appear as though it is in infinite free space [40]. Usually, to calculate the electric fields, HFSS can use the Finite Element Method (FEM). In this procedure, the solution space is discretized into tetrahedral finite elements where the E field can be approximated by a function and then substituted into the differential forms of Maxwell's equations [41]. Since the electric fields cannot be calculated out to infinity, a boundary condition can be applied. An ABC or PML can be used and each have their strengths. On the other hand, HFSS-IE offers the integral equation (IE) procedure which uses the Method of Moments (MoM) and is optimal for simulating large structures that are mostly conducting; however, it can result in a dense matrix that is not as efficient to calculate [41].

The FE-BI method is a combination of FEM and IE solution techniques and allows accurate solutions while eliminating large volume dimensions and thus, cuts down significantly on memory requirements [41]. Using FE-BI is more computationally heavy than ABC and PML, but it can be offset by using an airbox with smaller volume than ABC and PML [40]. Slide 17 in [40] shows a table that compares the radiating boundary

conditions summary for ABC, PML, and FE-BI, and goes into more detail describing different characteristics of FE-BI.

Since the horn simulation has a vast amount of free space between the source and the area of interest, it can be cut down into two separate air volumes and simulated using HFSS with the FE-BI.

3.2.4 Horn Setup with FE-BI Method 2

In order to simulate the horn, two FE-BI boxes were used. One went around the horn, and the other was located at the origin around the empty space that the aperture would sit in. Figure 21 below shows the set up. Again, the horn has been inserted as a 3D component and on another off-axis coordinate system to easily achieve off-axis excitation of the array without having to rotate and move the horn from the origin.

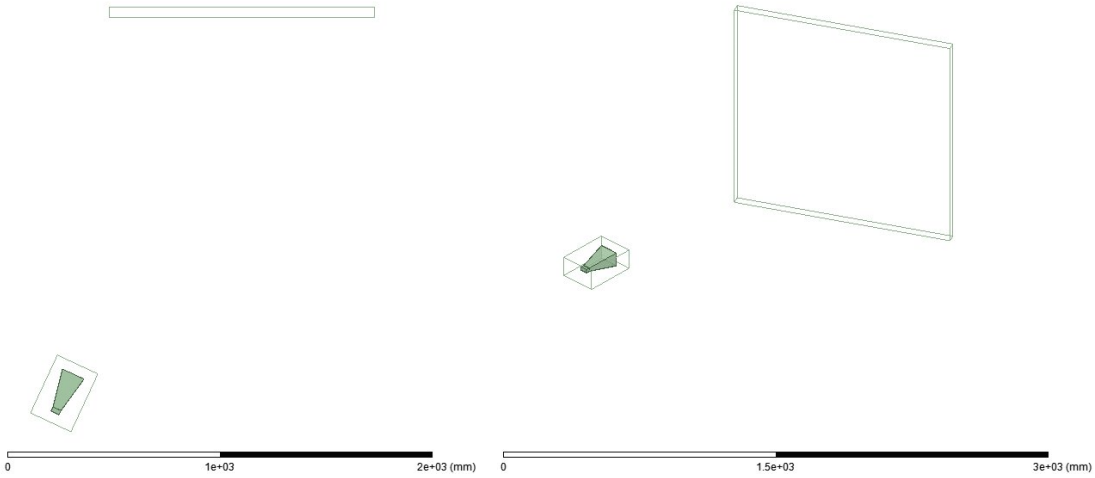


Figure 21 – Horn Reference Wave Simulation Setup 2

Once the set-up is made, simply assign the air boxes to a Hybrid FE-BI. Running the simulation was quick, only taking about 11 GB of RAM as opposed to more than 128 GB in the "brute force" method.

Once simulated, the E field Vector data was extracted using the Field Calculator. To make things simple, the center points of each patch were extracted as a file. On a side note, a plane could not be extracted, so two points in the *X* direction were extracted, but when processing in MATLAB, half of them were deleted since only *X* values of zero are needed. Once the data was extracted it was imported into MATLAB. The extracted data is listed in separate columns with the first three columns as the *X, Y,* and *Z* coordinates followed by the corresponding E field vector data. The E field vector data is in six columns with the columns in the following order: $E_{x_{Re}}$, $E_{x_{Im}}$, $E_{y_{Re}}$, $E_{y_{Im}}$, $E_{z_{Re}}$, and $E_{z_{Im}}$. The following Equations 3 and 4 are used to find the Electric field magnitude and phase from the E vector extracted from HFSS.

$$E_{phase} = \text{atan2d}\left(\frac{E_{z_{Im}}}{E_{z_{Re}}}\right) \tag{3}$$

$$E_{mag} = \sqrt{E_{z_{Re}}^2 + E_{z_{Im}}^2} \tag{4}$$

Using Equations 3 and 4, replacing $E_{z_{Re}}$ and $E_{z_{Im}}$ with the corresponding *X* and *Y* electric field components produces the following four graphs of the magnitude and the phase at the aperture in Figure 22.

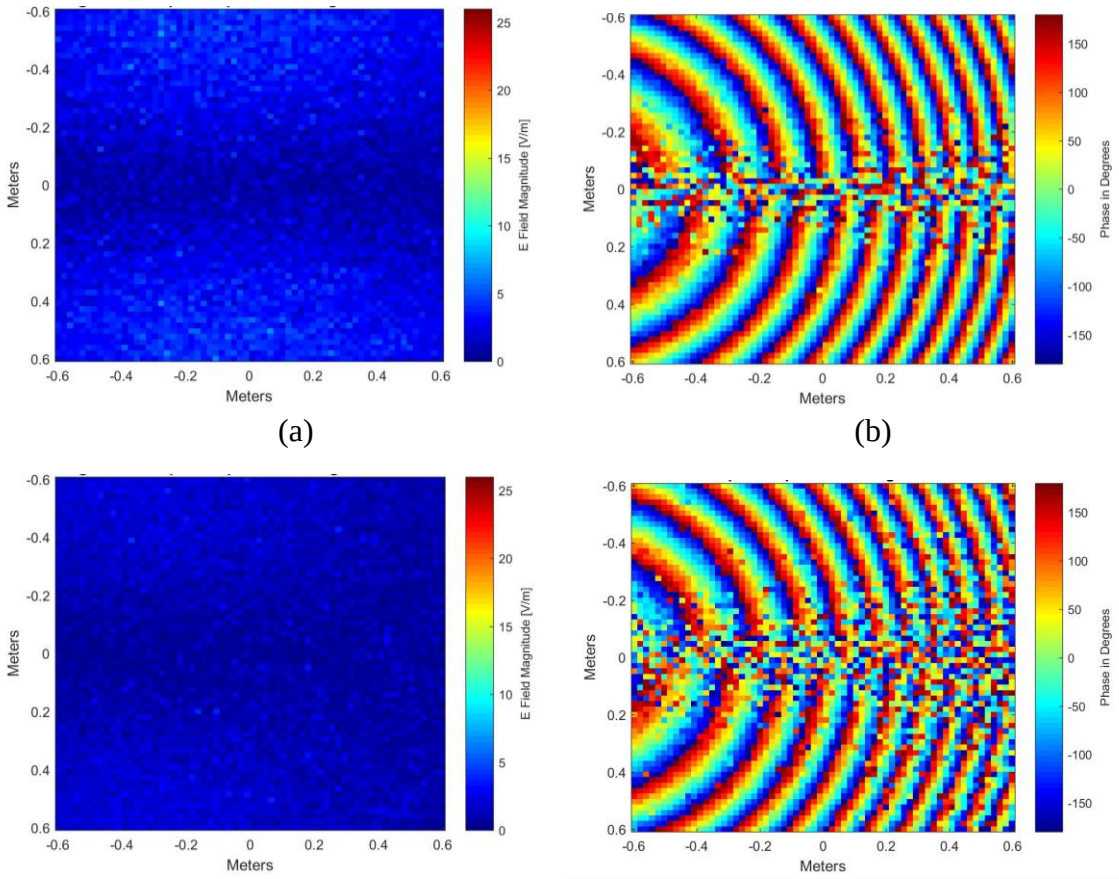
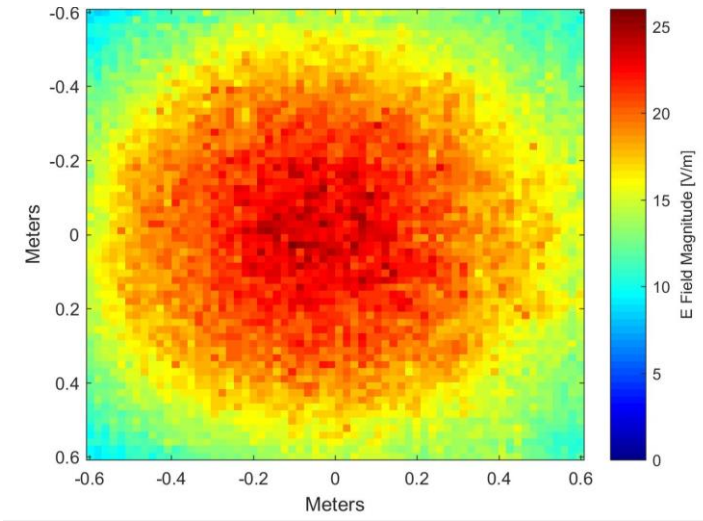


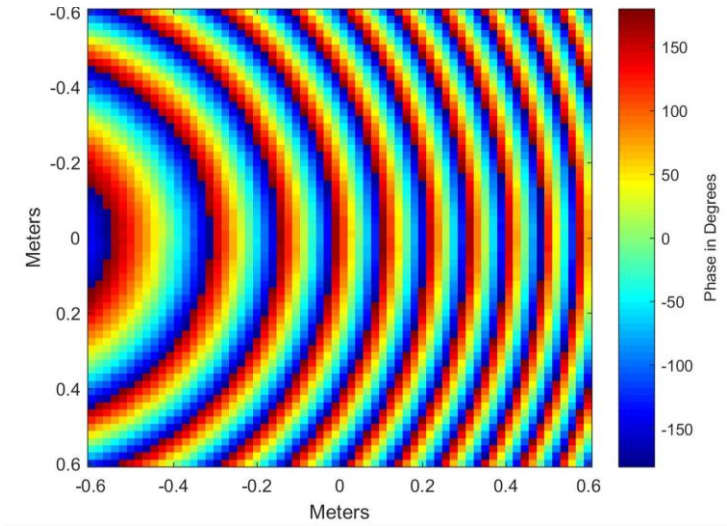
Figure 22 – *X* and *Y* Magnitude and Phase. (a) *X* Magnitude, (b) *X* Phase, (c) *Y* Magnitude, (d) *Y* Phase

In Figure 22 (a) and (c), the magnitudes of the electric fields show that the *X* and *Y* components of the electric fields do not contribute much, and it is mostly the *Z*

component that matters. As a result, the *Z* component phase will be used instead of extracting the overall phase of the complex electric field vector, which is more complex. Figure 23 shows the E field *Z* component using Equations 3 and 4.



(a)



(b)

Figure 23 – *Z* Magnitude and Phase

As shown in Figure 23 (b), the phase from the reference horn is very similar to Figure 2 (D) in [29]. The phase at each point at the input of the aperture has been simulated, and now the phase change needed at the aperture can be calculated in the next section.

3.3 Patch Size Calculation

All the information to make the hologram was simulated and calculated once the reference wave phase pattern was obtained from the horn. The next step to create the hologram was to use the fact that $\tilde{E}_A = \tilde{H} \cdot \tilde{E}_R$, where $\tilde{H}$ is the hologram Electric field, $\tilde{E}_A$ is the Electric field at the output of the aperture, and $\tilde{E}_R$ is the Electric field of the horn reference wave. Re-arranging that equation results in Equation 2, $\tilde{H} = \tilde{E}_A \cdot \tilde{E}_R^{-1}$, which is the equation needed to calculate the necessary phase change. The following Equations lead to Equation 5, the final equation that shows the hologram phase needed to yield the correct output aperture phase.

$$\tilde{H} = \tilde{E}_A / \tilde{E}_R$$

$$\tilde{H} = \frac{E_A \exp(j\Phi_A)}{E_R \exp(j\Phi_R)}$$

$$\tilde{H} = \frac{E_A}{E_R} \exp(j[\Phi_A - \Phi_R])$$

$$\Phi_H = \Phi_A - \Phi_R \tag{5}$$

Equation 5 shows that the hologram phase is simply a subtraction of the horn reference wave (the Z phase, Φ_R) from the phase needed at the output of the aperture (Φ_A which was calculated in Section 3.1). Figure 24 below shows the subtraction of the two phases and the resulting hologram phase.

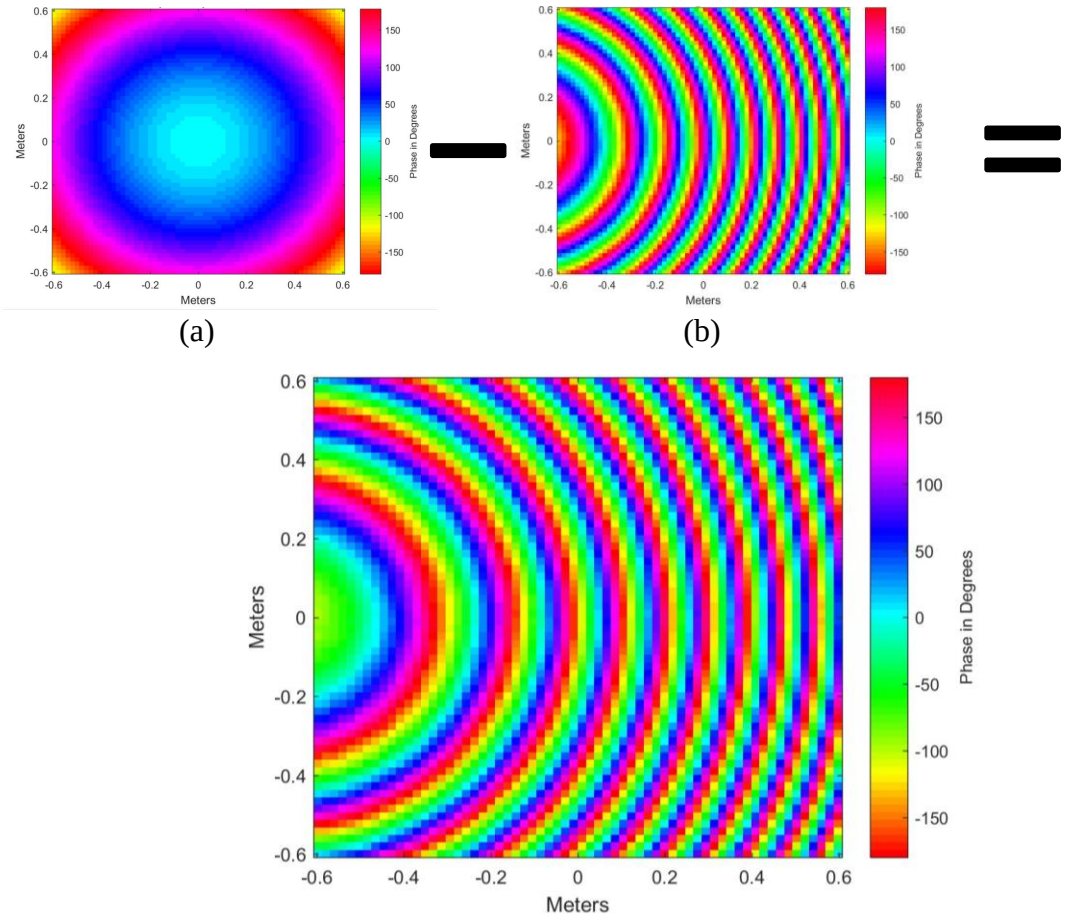


Figure 24 – Hologram Phase. (a) Φ_A, (b) Φ_R, and (c) Φ_H

Figure 24 (c) shows the phase that is needed at each point along the hologram. Again, the periodicity is 1.9 cm, so each pixel in the figure shows a 1.9 cm by 1.9 cm square. From the earlier simulation of circular patches with the Floquet Ports, it was shown that a full -180 to 180 phase range was not obtainable with the given substrates and thicknesses. As a result, the full hologram array phase range is not achievable. Figure 25 shows the hologram phase with the limitations of the physical patch.

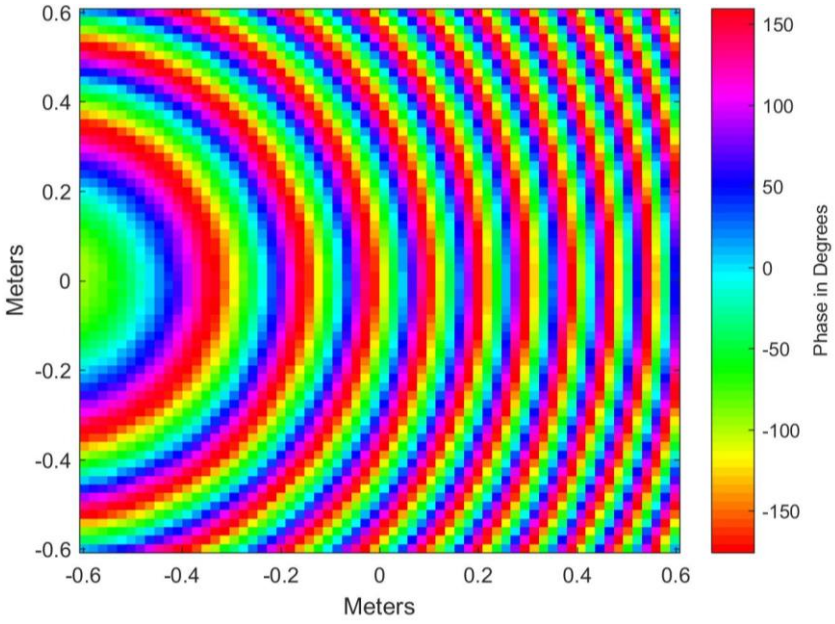


Figure 25 – Limited Hologram Phase

When examining Figure 25, it shows small change from the original hologram, which leads to the conclusion that the final hologram array will still work relatively well with a limited phase range. [29] has also shown with a more limited range that the hologram reflectarray will work.

To find out what patch size is needed at each location a MATLAB function was made that looks at every position on the array, and then it finds the closest match from the phases in the reflection response in Figure 14. The patch size needed at each location is plotted in Figure 26 below.

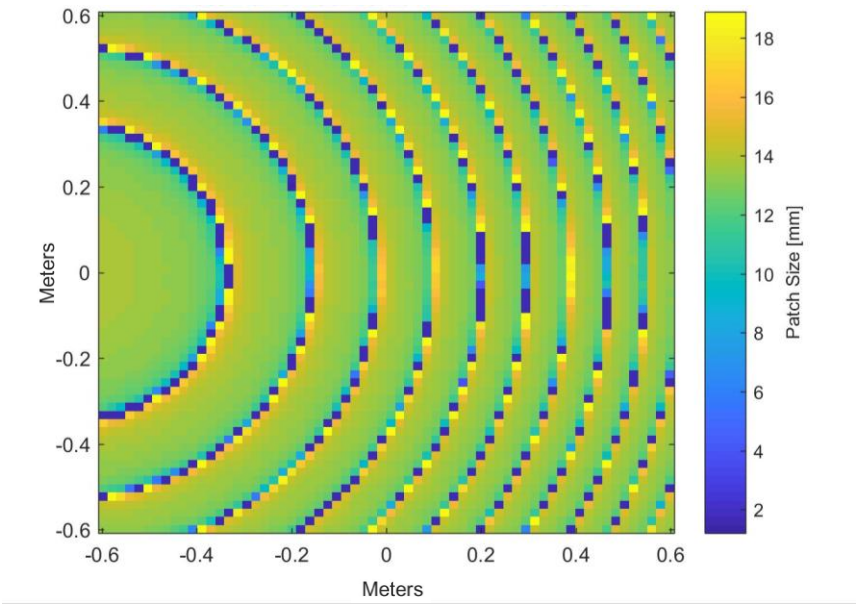


Figure 26 – Patch Size Needed at Each Pixel

From Figure 26, one can still see the general pattern is still there from Figure 24 and Figure 25. The circular patches range in size from 1.27 mm to 18.89 mm in diameter. Because the patches can only provide a certain phase range, the phase error has been plotted in the following Figure 27.

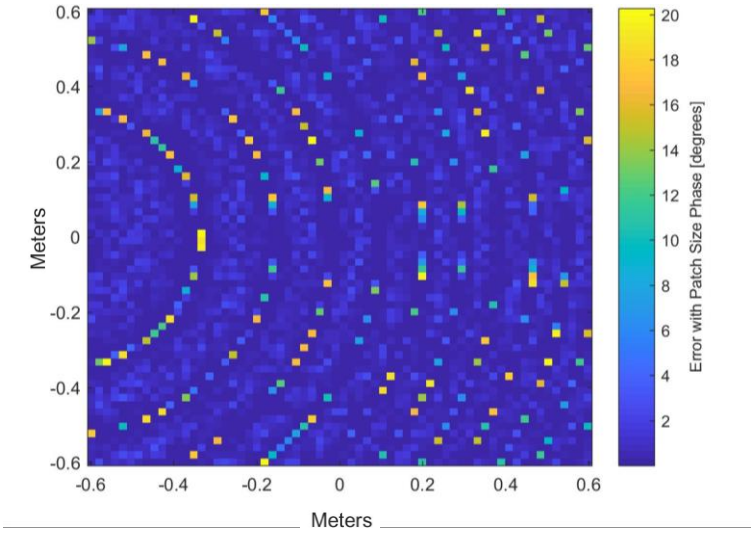


Figure 27 – Phase Error at each Pixel

The largest phase error that exists in Figure 27 is 20.28 degrees. Even though 20 degrees might seem large, it is actually smaller than the phase error in [29] that successfully simulated and built a holographic array. The following CHAPTER 4 Holographic Array will also prove that beam collimation is still possible.

CHAPTER 4. HOLOGRAPHIC ARRAY

4.1 HFSS Scripting

Now that the patch size at each location on the array has been determined, it must be implemented in HFSS to simulate. Since the array has 4096 patches, it would be very tedious to do all of this by hand. Thus, an HFSS script was made in MATLAB to run in HFSS. To first learn how to make a script, the record script to file under the tools section is very useful. This will allow the user to use the HFSS GUI and it will record everything done in the GUI to a script that accomplishes the same thing. For example, once the "Record Script To File" has been clicked and the filename saved, the user can draw an object or make a variable. Once the user has finished drawing or making some basic objects, the "Stop Script Recording" button under the Tools can be clicked to stop the recording. The script can be recorded in either in a python .py file or a .vbs file. For this project, the .py file format was used. A great resource to get started in creating an HFSS script is [42].

Once the script is recorded, it can be viewed in a text editor such as Notepad ++, and then a MATLAB script can be made to print to a .py file in the same format as the recorded script. Functions including createLocalVariable, createBox, and createCylinder were made in MATLAB that print out the formatted text that HFSS .py scripts need. The functions and the parent function are shown in Appendix A.3.2.4.

Once the script is made, it can be uploaded in HFSS. In a new project, the script was run in HFSS. On a side note, this might take about an hour and a half. Once it has run, a holographic array has been made in HFSS, and it can be seen in Figure 28 below.

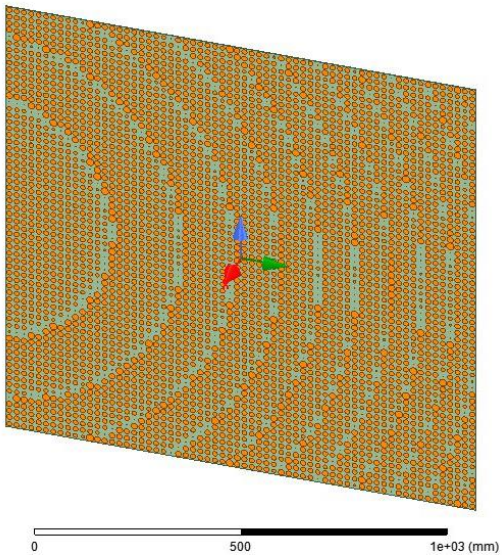


Figure 28 – 1.216-meter by 1.216-meter Holographic Array

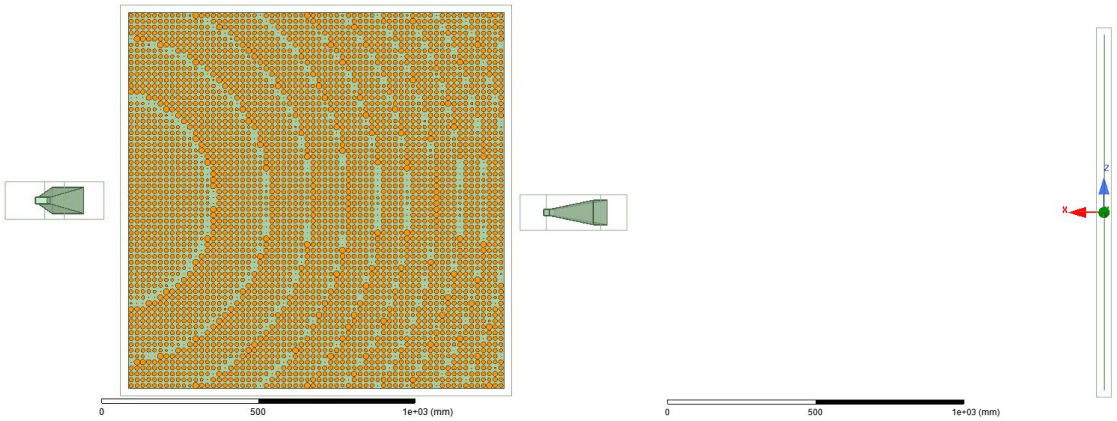
The holographic array was then exported as a 3D component so it can be easily inserted into another HFSS file for the final simulation.

4.2 HFSS Holographic Array Setup and Simulation

4.2.1 Holographic Array Simulation Setup

In a new project file, the 3D component horn was inserted into a new coordinate system to achieve the off-axis excitation, and the 3D component holographic array was inserted into the normal coordinate system at the origin. As was done in the horn reference wave simulation, two air boxes were formed around the horn and the holographic array and

assigned the hybrid FE-BI to make the simulation much faster. A picture of this can be seen in Figure 29.



(a) (b)

Figure 29 – Holographic Array and Horn Simulation Setup. (a) Front View, (b) Side Right View

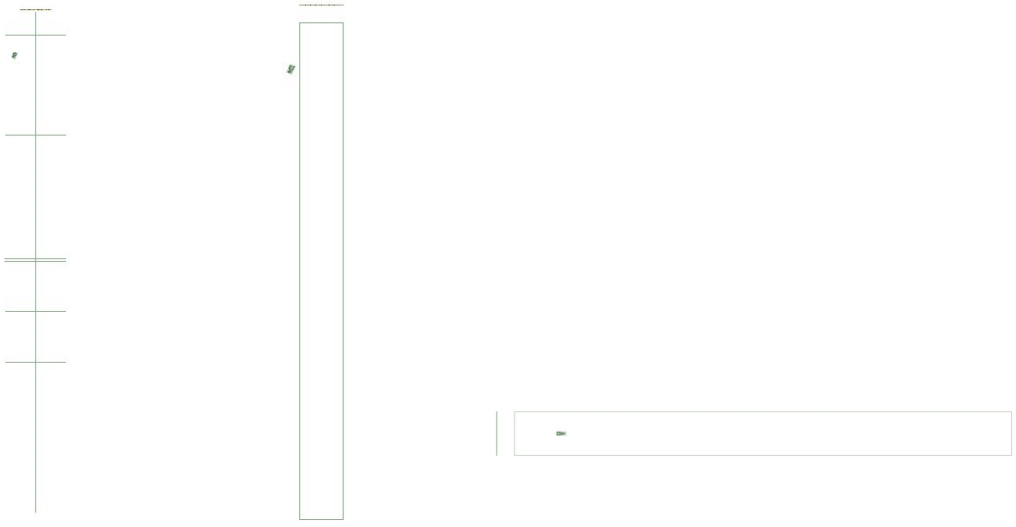
Figure 29 shows the horn and holographic array view from the front in (a) and from the right in (b). The horn is 25 degrees off the *X*-axis and located 190 cm away from the aperture. The box size pictured is larger than in the real simulation so that it can be seen more easily. In HFSS, the box size is $\frac{\lambda}{10}$ away from the objects, which is the optimal distance for FE-BI according to testing [41].

To see the results of the beam collimation, other features need to be added to the simulation. For a quicker simulation, near-field lines were added to view the collimation at the $Z = 0$ plane. The line locations can be seen in the following Table 4.

Table 4 – Near Field Lines

Line and Description	Distance From the Aperture (meter)
Line 1: Near the aperture	1
Line 2: Halfway between the focal point and aperture	5
Line 3: Near the focal point	9.9
Line 4: At the focal point	10
Line 5: Farther from the focal point	12
Line 6: Even farther from the focal point	14
Line 7: Long Line from Aperture to 20 Meters at $Y = 0$ and $Z = 0$	10 cm – 20 meters

Once the near-field lines are inserted the setup will look like Figure 30 (a). For a better visual, but longer simulation, a very long thin box was inserted in the *XY* plane as shown in Figure 30 (b) and in the *XZ* plane as shown in Figure 30 (c).



(a) (b) (c)

Figure 30 – Final Holographic Array Simulation Setup

The air boxes are the width of the aperture, 14 meters long, and the width was 0.5 mm or 1 mm to shorten the simulation time. Once the simulation setups were complete, they were

sent to simulate on the PACE cluster, which can accommodate a large amount of RAM (greater than the desktop has which is 32 GB). The simulation in Figure 30 (a) took about 20 hours with 62 GB RAM allocated, Figure 30 (b) took more than 60 hours with 128 GB RAM allocated, and Figure 30 (c) took more than 60 hours with 128 GB RAM allocated.

4.2.2 Simulation Results

Once the simulations finished, the results were exported back from the cluster and viewed. Figure 31 below shows the magnitude of the electric field results from Lines 1-6 in Table 4 and in Figure 30 (a).

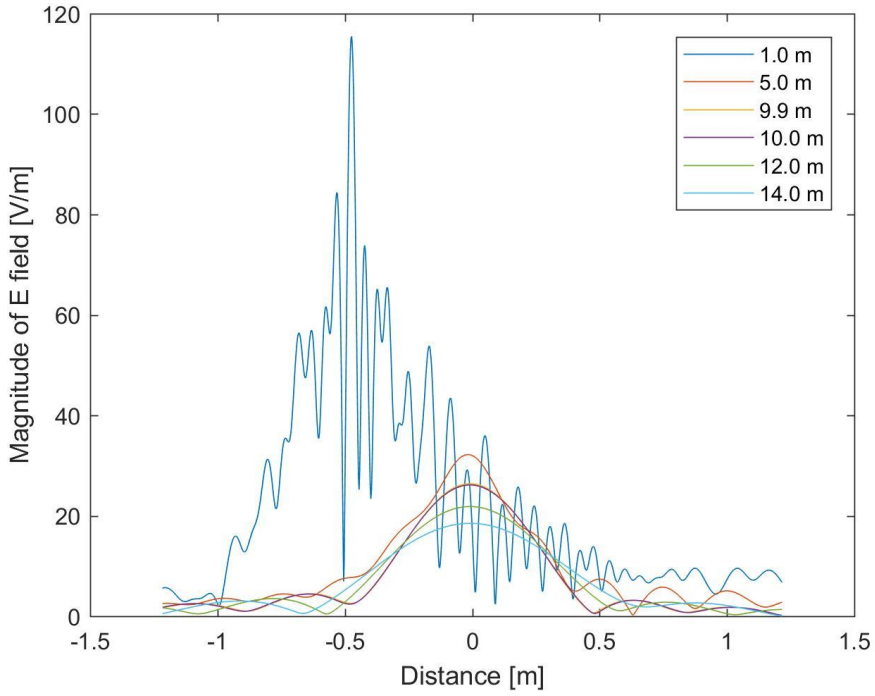


Figure 31 – Near Field Lines 1-meter, 5-meter, 9.9-meter, 10-meter, 12-meter, and 14-meter

Figure 31 shows the beam collimation from the aperture, to the focal point, and away from the focal point. Each coloured line is a different distance from the aperture, with the

purple line being the focal point. The closest 1-meter line has a large amplitude partially because of the horn's interference since it is so close to the aperture.

The magnitude of the electric field along Line 7 from Table 4 and Figure 30 (a) is plotted in Figure 32 below.

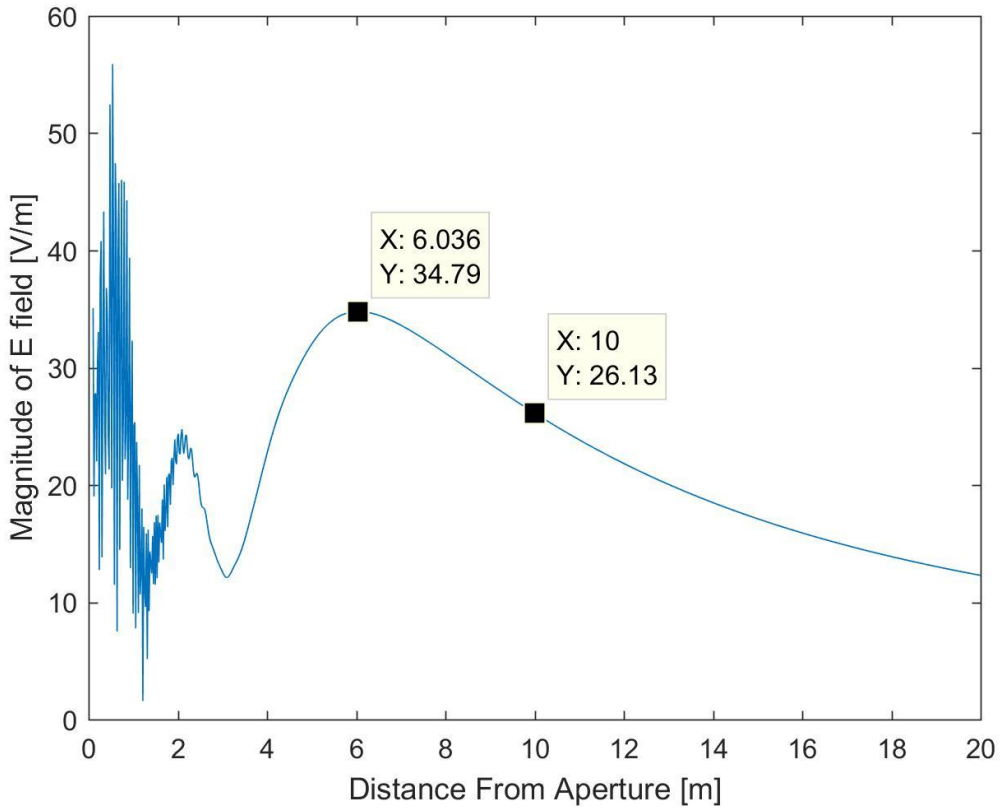
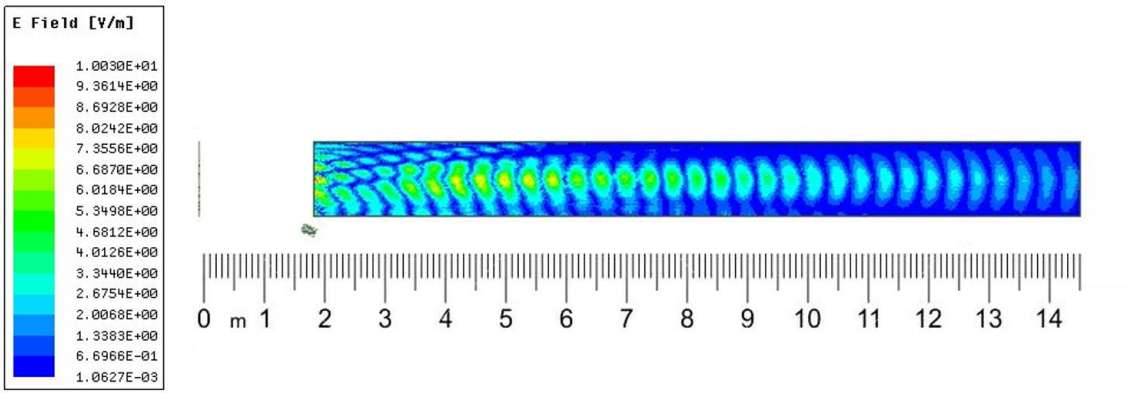


Figure 32 – Magnitude of the Electric Field along Line 7

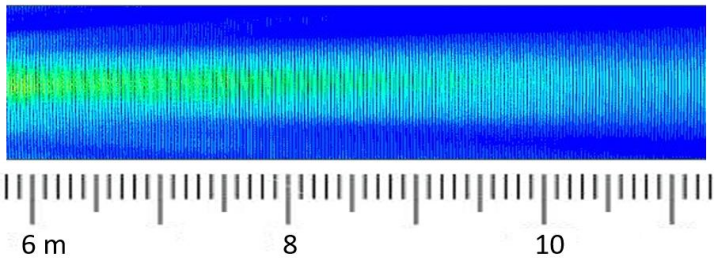
Although the focal point is at 10 meters, it is not the highest point of the electric field. The peak electric field is instead between the aperture and the focal point at 6.036 meters from the aperture at 34.79 V/m. This is due to the field spreading factor $\frac{1}{R}$ [22]. From [22] there is a parameter γ which shows the focal distance normalized to the far-field's region

boundary, where $\gamma = \frac{F}{\frac{2L^2}{\lambda}}$. For very small values of γ, this shift disappears; however, as γ approaches 1, the peak power slowly approaches a value corresponding to a far field focused array [22]. For the array simulated above, γ = 0.1749. To make the focal point occur closer to the peak power, γ must be very small.

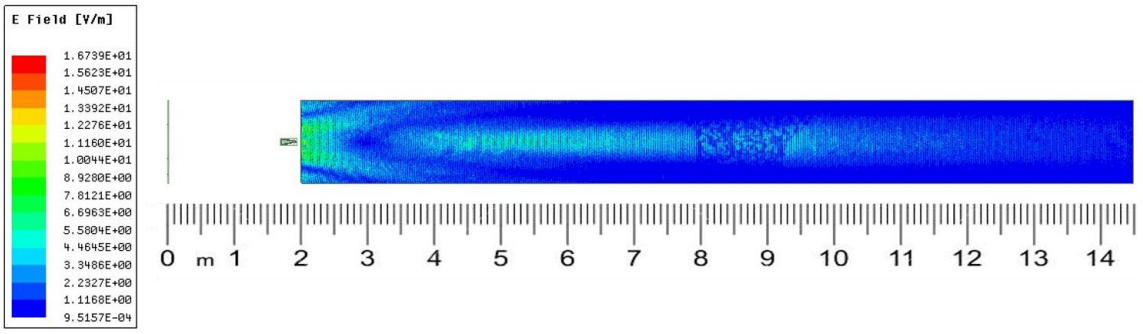
To visualize the beam collimation more, Figure 30 (b) and (c) show a setup with a very thin air box. Once the simulation was complete, a non-model sheet was inserted, and the magnitude of the electric field was plotted on the sheet. The results are in the following three plots in Figure 33.



(a)



(b)



(c)

Figure 33 – Magnitude of Electric Field. (a) *XY* plane, (b) Zoomed in *XY* plane, and (c) *XZ* plane

Although in Figure 33 (a) it looks weirdly pulsed, it does not look that way when magnified, as seen in Figure 33 (b). Since the horn's signal is so much larger, the electric field was not plotted in front of the horn because the horn interference made the beam collimation disappear. Figure 33 shows the beam collimation and the spreading of the signal after.

4.2.3 Other Array Simulation Results

Since a MATLAB script was written to construct a hologram, it was simple to switch and or replace some lines of code to make a hologram out of Rogers substrate or make the array smaller. All the following arrays have the same patch periodicity of 1.9 cm, have an even number of patches, and are square. The Rogers substrate hologram was made to be the same size, 1.216-meter by 1.216-meter with a focal point at 10 meters. The same procedure was followed including running a simulation to get the reflection response in a smaller step size of 0.01 mm and running an HFSS script to create the hologram in HFSS. Section 2.2.5 showed that the Rogers substrate had a more suitable reflection response with the magnitude not dipping as far, and the phase curve less steep than the FR4 reflection response curve. The results below will show how this affects the beam collimation. Figure

34 shows the hologram with the Rogers substrate, and Figure 35 and Figure 36 show the results. The same setup was used from the FR4 hologram simulation except for the *XZ* air box.

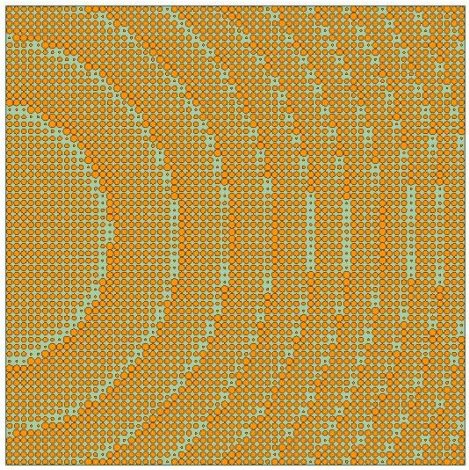


Figure 34 – Hologram with Rogers Substrate

This hologram looks identical to the FR4 with the naked human eye; however, the patch sizes are different. Figure 35 below shows the magnitude of the electric field over the near field lines.

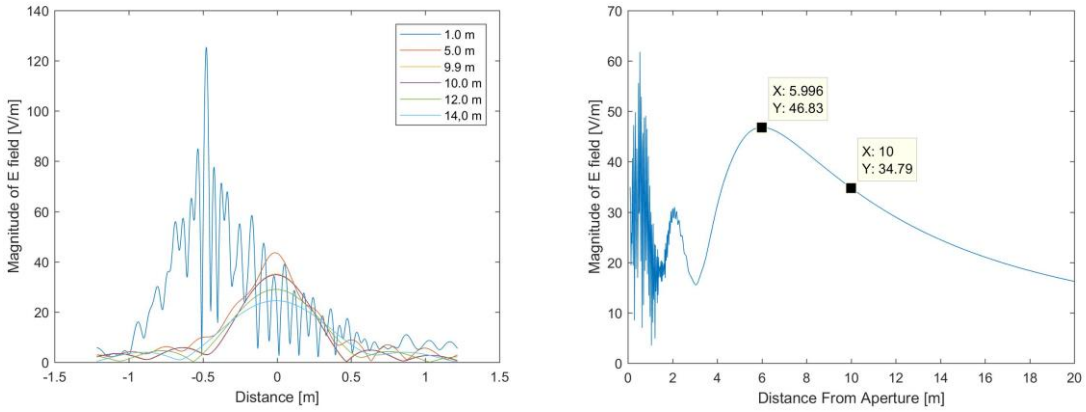


Figure 35 – Magnitude of the Electric Field. (a) Lines parallel to y-axis at different distances away and (b) One long line from 10 cm to 20 meters

Figure 36 below shows the electric field over a non-model sheet in the *XY* plane.

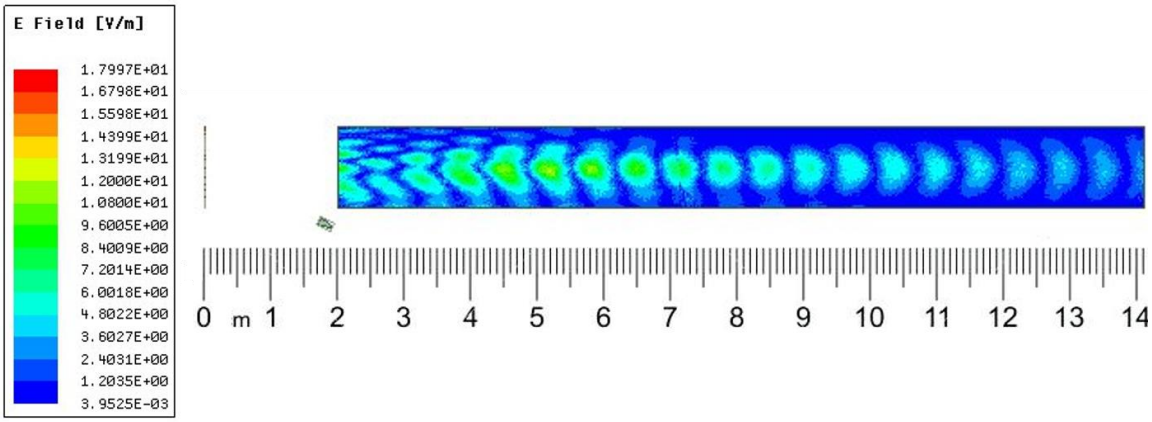


Figure 36 – Magnitude of Electric Field along Sheet in *XY* Plane

Though the above Figure 36 also shows a pulse pattern, when magnified, the pulse gone and it looks similar to Figure 33 (b). The Rogers beam collimation had a higher power output. The electric field was about 10 V/m higher at the max electric field point and at the focal point. Figure 37 below shows a comparison of the FR4 hologram and the Rogers hologram.

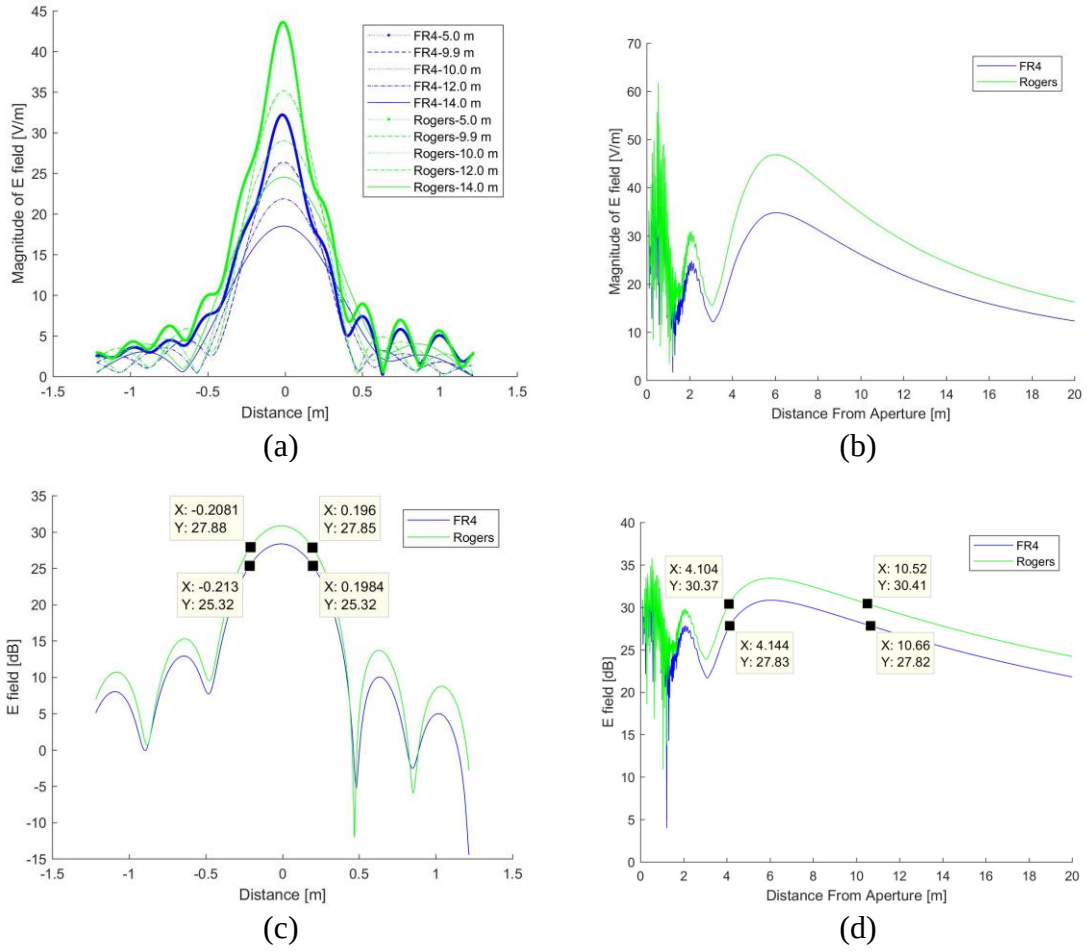


Figure 37 – FR4 and Roger Hologram Magnitude of the Electric Field Comparison. (a) Magnitude of the electric field at lines 1-6, (b) Magnitude of the electric field of line 7, (c) Electric field in dB of line 4, and (d) Electric field in dB of line 7. Lines are from Table 4

The beam collimation has a higher power in the Rogers substrate, which is probably a result of the magnitude of the reflection coefficient dropping less. Since the Rogers reflection response curve was shifted to the right slightly, the range was slightly smaller than the FR4 phase range; however, this does not seem to be a problem in the beam collimation. The 3-dB width of the beam is about the same for both. The points on Figure 37 (c) and (d) are about 3 dB down from the maximum value. If price is not an issue,

Rogers might prove to be a better choice for fabrication if the machine has limitations on how precise it can be.

Two other holograms were designed and simulated on FR4 substrate but in smaller sizes. The first was a 0.456-meter by 0.456-meter aperture with a focal point at 3 meters. Figure 38 below shows the hologram design in HFSS.

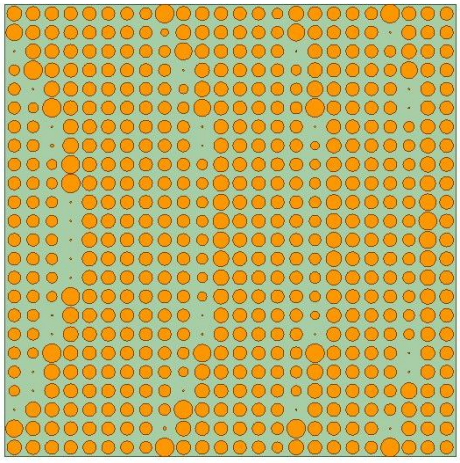


Figure 38 – 0.456-meter by 0.456-meter Hologram

This aperture has 24 by 24 patches. Since the size is much smaller, it would be an easier aperture to fabricate. The 0.456-meter aperture was simulated using a substrate thickness of 59 mils since the FR4 available in the laboratory is 59 mils. The following plots in Figure 39 are made using a 20-cm, 1.5-meter, 3-meter, 4-meter, 5-meter, and 1-cm to 10-meter line and an *XY* non-model sheet.

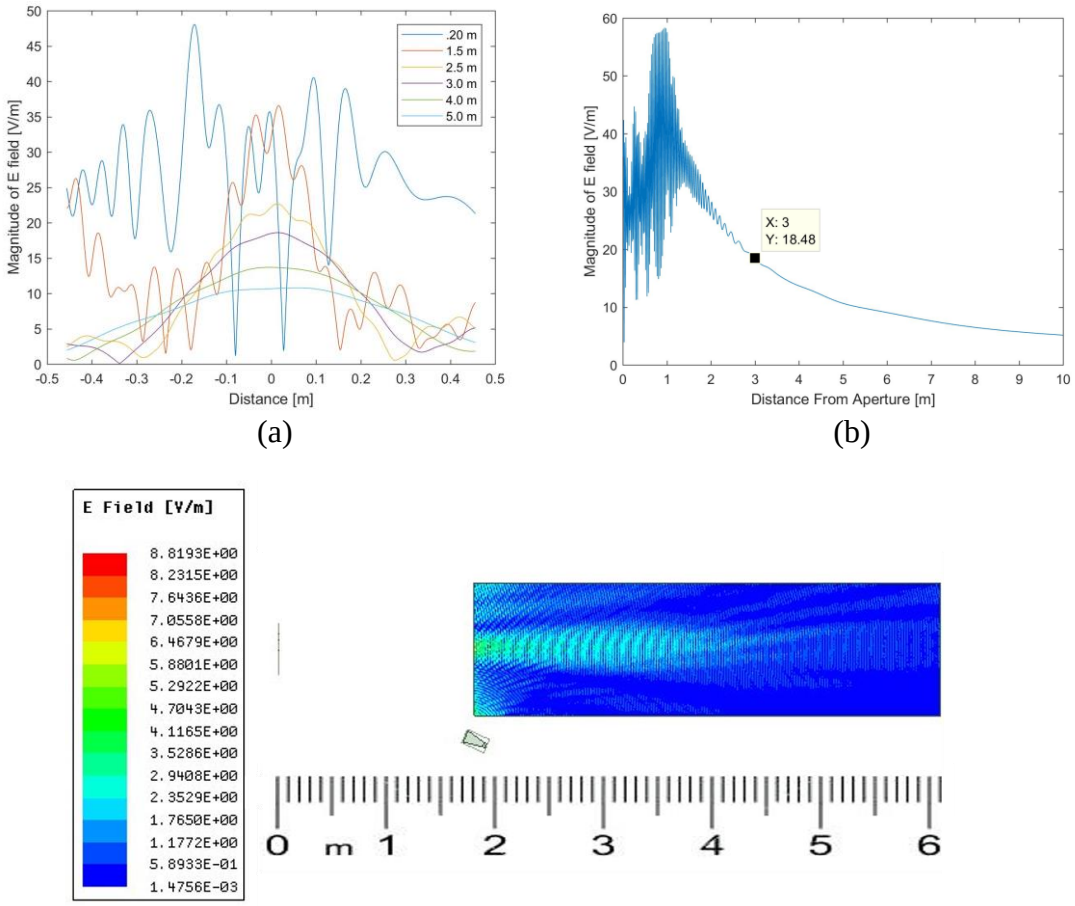


Figure 39 – Magnitude of the Electric Field for 0.456-meter Aperture. (a) Field along 6 lines, (b) Field along one long line, and (c) Field on non-model sheet

Figure 39 (a) and (c) shows the beam collimation is still present. However, there is interference from the horn that overlaps the peak electric field as seen in Figure 39 (b), and it also interferes in line 1.5-meters as well as in the 20-cm line shown in Figure 39 (c). In this 0.456-meter aperture, γ = 0.373, which is higher than in the 1.216-meter aperture (γ = 0.1749). It is difficult to really see how the larger γ affects the gap between the peak electric field and the focal point. In future simulations, it might be useful to move the horn closer to the aperture if the aperture is only half a meter and pick a closer focal point.

The second smaller aperture that was designed and simulated was a 0.912-meter by 0.912-meter aperture with a focal point at 5 meters. Figure 40 below shows the hologram in HFSS.

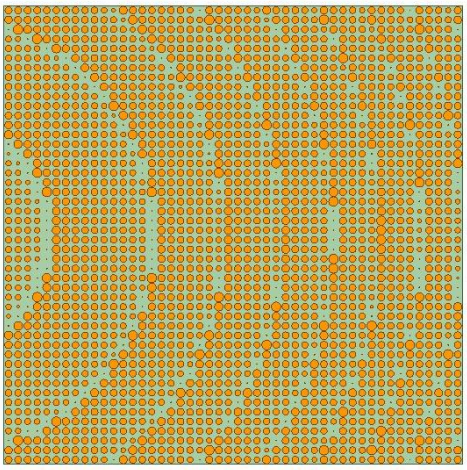
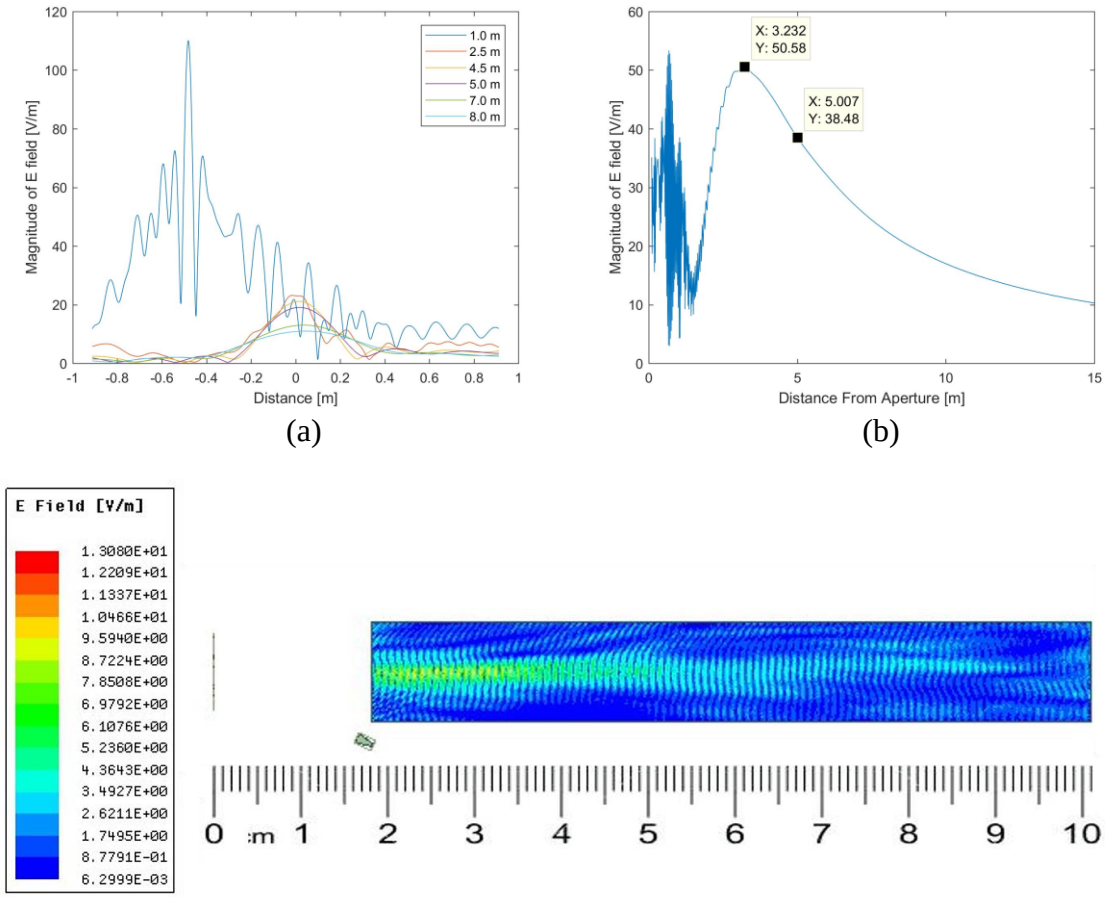


Figure 40 – 0.912-meter by 0.912-meter Hologram

The hologram is made of a grid of 48 by 48 patches, totaling to 2304 patches. Again, it was simulated with a substrate thickness of 59 mils using FR4. The following plots of the magnitude of the electric field in Figure 41 were made using a 1-meter, 2.5-meter, 4.5-meter, 5-meter, 7-meter, 8-meter, and 10-cm to 15-meter lines and a non-model sheet in the *XY*-plane.



(a) (b)

(c)

Figure 41 – Magnitude of the Electric Field for 0.912-meter Aperture. (a) Field along 6 lines, (b) Field along one long line, and (c) Field on non-model sheet

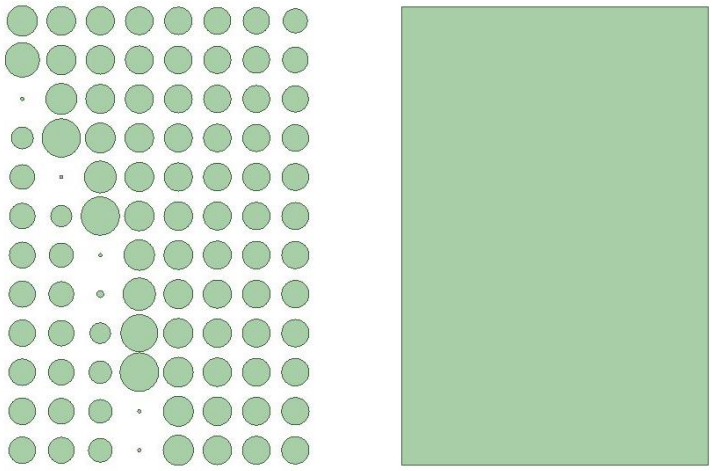
For the 0.912-meter aperture, γ = 0.155 which is smaller than the 1.216-meter aperture ($\gamma = 0.1749$). In Figure 41 (c), the focal point is 1.768 meters from the peak electric field point whereas in the 1.216-meter, the focal point is about 4 meters away from the peak electric field point in both the FR4 and the Rogers holograms in Figure 32 and Figure 35 (b) respectively. Figure 41 (a) and (c) show that the beam collimation is again successful.

4.3 Starting Fabrication

Since 0.456-meter aperture is the smallest, it is the easiest to fabricate. However, it is still relatively large, so it needs to be made on six separate FR4 boards and then pieced together. The outline of how the array will be divided up is shown in Figure 42 below.

Figure 42 – 0.456-meter Aperture Fabrication Layout

In order to fabricate six separate pieces, the HFSS design was cut into 6 separate sections and put in separate HFSS design files, and then each segment was centered about the origin and put into the *XY* plane. Each patch within one section was added together to make one single object. In addition, an "outline" was made that is the same size as one of the parts. This will be used to determine how to cut out the section from the FR4 board. Figure 43 below displays the Top Left section and the outline that was made.



(a) (b)

Figure 43 – Export Setup

Since HFSS does not export Gerber files easily, the face of each object (all the patches within a section or an outline) was selected, and then each object was exported as a .dxf file. In total there were 7 separate .dxf files: Top Left, Top Middle, Top Right, Bottom Left, Bottom Middle, Bottom Right, and Outline. Gerber files can easily be exported from Keysight ADS, so the .dxf files were imported into Keysight ADS and then exported as Gerber files to a flash drive. Once the array is in 7 separate Gerber files on a flash drive, it can be fabricated on a machine. Currently, the correct settings on the laser milling machine have yet to be determined to correctly fabricate a board with FR4 substrate. There are currently issues with the machine not removing all the copper from in between patches on the first run as shown in Figure 44 below.

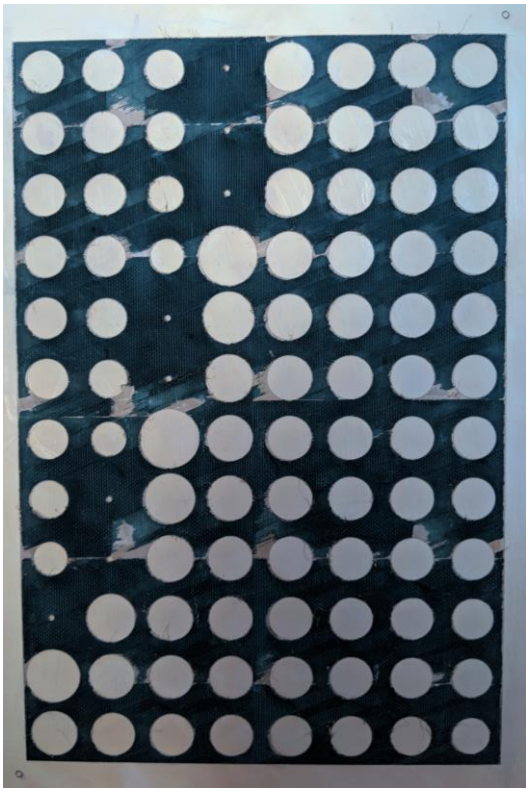


Figure 44 – Fabricated Bottom Left Board

Figure 44 shows an attempt at fabricating the Bottom Left section of the 0.456-meter aperture. This picture was taken after scraping off a lot of the easily lifted copper that still remained in between patches. The fabrication of this section took about 4 hours, and it will still need to be run through the milling process again to remove the remaining unwanted copper. Fiducials have been made so that the board can be run again through the laser milling machine.

CHAPTER 5. CONCLUSIONS AND FUTURE WORK

A holographic transmit reflect array at 5.8 GHz for wireless power transfer in the Fresnel-zone was successfully designed and simulated. A few designs have been made including two 1.216-meter apertures made from 4096 patches on FR4 and Rogers 60 mils and two smaller 0.456-meter and 0.912-meter holographic arrays made up of 576 and 2304 circular patches respectively on FR4 59 mils. The Rogers substrate yielded a better power reflection than the FR4 substrate in simulation, but both seem to collimate the beam equally well. Different sized apertures can be designed with successful collimation, but the distance of the focal point should be chosen carefully; the ratio of the focal point to the upper boundary of the Fresnel-zone should be kept small to have the focal point and the peak reflected power point near or in the same spot.

Circular patches successfully create a collimated beam, and they have the potential to make the array polarization-independent. Since the axial ratio is understandably high as the scan angle approaches 90 degrees, the angle of the incident wave should be kept lower if circular polarization is desired. Future research could explore how polarization-independent the aperture truly is and fabricate and test the array designed in this thesis. When testing the array with the magnetron as shown in Figure 2, it is important to take precautions, since the power output may be high. Refer to FCC Policy on Human Exposure for maximum permissible exposure limits.

Future research could also investigate how well the fabricated FR4 array works in comparison to the simulated one. There is slight concern about how precise the laser milling machine can be, and whether it can fabricate the patches precisely enough to

successfully collimate a beam. Perhaps using a different substrate that has a better phase response could be explored as well.

The successful collimation of a beam from a reflect array is dependent on many variables staying constant. Since the phase response varies based on the incident angle, shape of the patch, distance of the Floquet port, and substrate type the patch is on, a simulation of the phase reflection response with changing patch size needs to be completed with the deembed setting on and by specifying the desired shape of the patch, incident angle, and substrate type. This graph of reflected phase versus patch size is very important in the creation of the hologram. The incident wave pattern's phase is also taken exactly where the aperture is located and is used to create the final hologram, so the distance and type of antenna used must be constant. Finally, the apertures designed use an equation that only yields a focal point at a fixed distance away from the aperture's center.

The designed and simulated holographic array from this paper has limited applications without active elements. Future research can explore how to make the array dynamic so a few variables can be changed, such as the use of varactor diodes as mentioned before. Once dynamic, reflect arrays have the potential to play critical roles in applications such as space solar power and mid-air drone charging.

APPENDIX A. MATLAB CODE

A.1 Phase Distribution at the Output of the Aperture

```
%% Graph the phase distribution needed at the output of the
aperture
% Emily Backer
% Phase Calculations - Phase needed at the output of the
aperture
% Code needs even number of patches and assumes array is in
the yz plane
% Three main sections
% (1) single length and focal length calculation and figure
% (2) section will graph multiple figures with different
% aperture sizes. The Ls vector takes what aperture lengths
% to graph
% (3) animation of varying lengths

% clear slate
close all;          % close figure windows
clear variables;    % clear memory
clc;                % clear command window
fclose('all');      % close files

% Constants
f = 5.8e9; %5.8GHz
c = physconst('lightspeed'); %299792458
lambda = c/f;

% Variables
F = 10; %Distance to focal point
L = 1.216; %(1meter) %Aperture Size
FresnelRegionStart = 0.62*sqrt((L^3)/lambda); %start of
fresnel region
FresnelRegionEnd = 2*(L^2)/lambda; %Boundary for F, upper
bound
periodicity = 1.9e-2; %each patch periodicity of 1.9cm

%% (1) Calculations for single aperture
pixel = ceil(L/periodicity); %define how many
patches/pixels there will be
if mod(pixel,2)==1 %make pixel even number
    pixel = pixel-1;
end
```

```
Matrix = zeros(pixel,pixel);
A = periodicity/2;
y = floor(-
pixel/2)*periodicity+A:periodicity:floor(pixel/2)*periodici
ty-A;
z = floor(pixel/2)*periodicity-A:-periodicity:floor(-
pixel/2)*periodicity+A;

for r = 1:pixel
    for c = 1:pixel
        Matrix(r,c) = (2*pi/lambda)*(sqrt(z(r).^2 + y(c).^2
+ F^2)-F);
    end
end

    figure
    Unwrapped = Matrix;
    UnwrappedDeg = rad2deg(Matrix);
    Matrix = wrapTo180(UnwrappedDeg);
    imagesc(y,z,Matrix);
    xlabel('Meters');
    ylabel('Meters');
    colormap(hsv);
    h = colorbar;
    ylabel(h,'Phase in Degrees');
    title(sprintf('Phase Needed for Aperture Length = %0.3f
meters',L));
    set(gca,'YDir','normal') %Make the y axis go from - to
+


%% (2) Vary Size of Aperture and Graph multiple graphs
Ls = [0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7];
for i = 1:length(Ls)
    L = Ls(i); %length of aperture
    pixel = ceil(L/periodicity); %define how many
patches/pixels there will be
    if mod(pixel,2)==1 %make pixel even number
        pixel = pixel-1;
    end

    Matrix = zeros(pixel,pixel);
    % make the center of the patch the location for
calculating the phase
    % necessary
    A = periodicity/2;
```

```
    y = floor(-
pixel/2)*periodicity+A:periodicity:floor(pixel/2)*periodici
ty-A;
    z = floor(pixel/2)*periodicity-A:-periodicity:floor(-
pixel/2)*periodicity+A;

    % calculate the matrix phases
    for r = 1:pixel
        for c = 1:pixel
            Matrix(r,c) = (2*pi/lambda)*(sqrt(z(r).^2 +
y(c).^2 + F^2)-F);
        end
    end

    % make figure
    figure
    Unwrapped = rad2deg(Matrix);
    Matrix = wrapTo180(rad2deg(Matrix));
    imagesc(y,z,Matrix);
    xlabel('Meters');
    ylabel('Meters');
    colormap(hsv);
    h = colorbar;
    ylabel(h,'Phase in Degrees');
    title(sprintf('Phase Needed for Aperture Length = %0.2f
meters',Ls(i)));
    set(gca,'YDir','normal') %Make the y axis go from - to
+


end

%% (3) Calculations and Animation with varying Aperture
lengths
% to make an animation for varying focal distances,
% simply make a vector of Fdists and in the loop change Ls
% to Fdists and set F each time
hh = figure;
axis tight manual
filename =
'AnimatedVaryingLengthPhasePlotsStepSize0.05.gif';

StepSize = 0.05;
Ls = 0.5:StepSize:1.7;

for i = 1:length(Ls)
    L = Ls(i);
```

```
    pixel = ceil(L/periodicity); %define how many
patches/pixels there will be
    if mod(pixel,2)==1 %make pixel even number
        pixel = pixel-1;
    end

    % make the center of the patch the location for
calculating the phase
    % necessary
    Matrix = zeros(pixel,pixel);
    A = periodicity/2;
    y = floor(-
pixel/2)*periodicity+A:periodicity:floor(pixel/2)*periodici
ty-A;
    z = floor(pixel/2)*periodicity-A:-periodicity:floor(-
pixel/2)*periodicity+A;

    % calculate the matrix phases
    for r = 1:pixel
        for c = 1:pixel
            Matrix(r,c) = (2*pi/lambda)*(sqrt(z(r).^2 +
y(c).^2 + F^2)-F);
        end
    end

    Unwrapped = rad2deg(Matrix);
    Matrix = wrapTo180(rad2deg(Matrix));
    imagesc(Matrix);
    colormap(hsv);
    h = colorbar;
    ylabel(h,'Phase in Degrees');
    title(sprintf('Length of Aperture L = %0.3f
meters',Ls(i)));
    set(gca,'YDir','normal') %Make the y axis go from - to
+

    drawnow
    frame = getframe(hh);
    im = frame2im(frame);
    [imind,cm] = rgb2ind(im, 256);

    if i==1
        imwrite(imind,cm,filename, 'gif', 'Loopcount',
inf);
    else
        imwrite(imind, cm, filename, 'gif', 'WriteMode',
'append');
```

```
    end
end
```

A.2 Graphing the E field MATLAB

```
%Patch Antenna Test with Off Axis Illumination
%import HFSS file and extract vector information, then find
phase information, plot
%Plot E field X, E field Y, and E field Z

%% clear slate
close all;          % close figure windows
clear variables;    % clear memory
clc;                % clear command window
fclose('all');      % close files

%% Aperture Info
L = 1216; %100mm
Pixel = 19; %1mm
numZpos = L/Pixel; %number of z and y positions, (i.e. -
180.5 to 180.5 steps of 19)
numYpos = numZpos;

%% Read in file and Extract
PatchInfo = dlmread('HornReferenceWaveV3-1.fld'); %file
contains complex E vector data

%Extract x y z coordinates and corresponding phase
FirstHalf = 1:length(PatchInfo)/2';
x = PatchInfo(FirstHalf,1);
y = PatchInfo(FirstHalf,2);
z = PatchInfo(FirstHalf,3); %had to import x info at 0mm
and 1mm, but only need phase at 0mm
ExReal = PatchInfo(FirstHalf,4);
ExImag = PatchInfo(FirstHalf,5);
EyReal = PatchInfo(FirstHalf,6);
EyImag = PatchInfo(FirstHalf,7);
EzReal = PatchInfo(FirstHalf,8);
EzImag = PatchInfo(FirstHalf,9);

%% Calculate Phase and Magnitude
for AA = 1:3
    if AA==1
        EPhase = atan2d(ExImag,ExReal);
        EMag = sqrt(ExReal.^2+ExImag.^2);
```

```
    elseif AA==2
        EPhase = atan2d(EyImag,EyReal);
        EMag = sqrt(EyReal.^2+EyImag.^2);
    else
        EPhase = atan2d(EzImag,EzReal);
        EMag = sqrt(EzReal.^2+EzImag.^2);
    end
PatchPhaseMagnitude = EPhase;

%% Re-arrange into a Matrix for imagesc
% Z and Y increases from -20mm to 20mm, but Y increases
slower
% i.e. Y = -20mm, while Z -20mm:1mm:20mm, then Y = -19mm
and Z increases
% again -20mm:1mm:20mm

PhaseMatrix = zeros(numYpos,numZpos);
MagMatrix = zeros(numYpos,numZpos);
row = numYpos:-1:1;
col = 1:1:numZpos;
indexStart = 1;
indexStop = numZpos;
for c = col
    PhaseMatrix(row,c) =
PatchPhaseMagnitude(indexStart:indexStop);
    indexStart = indexStart + numZpos;
    indexStop = indexStop + numZpos;
end
indexStart = 1;
indexStop = numZpos;
for c = col
    MagMatrix(row,c) = EMag(indexStart:indexStop);
    indexStart = indexStart + numZpos;
    indexStop = indexStop + numZpos;
end
if AA==1
    PhaseXMatrix = PhaseMatrix;
    MagXMatrix = MagMatrix;
elseif AA==2
    PhaseYMatrix = PhaseMatrix;
    MagYMatrix = MagMatrix;

else
    PhaseZMatrix = PhaseMatrix;
    MagZMatrix = MagMatrix;
end
end
```

```
%% Plot
figure(1)
imagesc(y,z,PhaseXMatrix);
xlabel('Meters');
ylabel('Meters');
colormap(jet);
h = colorbar;
ylabel(h,'Phase in Degrees'); %E Field [V/m]
title(sprintf('X Phase at Input of Aperture of Length =
%0.2f mm',L)); %E field at...

figure(2)
imagesc(y,z,MagXMatrix,[0,26]);
xlabel('Meters');
ylabel('Meters');
colormap(jet);
h = colorbar;
ylabel(h,'E Field Magnitude [V/m]'); %E Field [V/m]
title(sprintf('X Magnitude at Input of Aperture of Length =
%0.2f mm',L)); %E field at...

figure(3)
imagesc(y,z,PhaseYMatrix);
xlabel('Meters');
ylabel('Meters');
colormap(jet);
h = colorbar;
ylabel(h,'Phase in Degrees');
title(sprintf('Y Phase at Input of Aperture of Length =
%0.2f mm',L));

figure(4)
imagesc(y,z,MagYMatrix,[0,26]);
xlabel('Meters');
ylabel('Meters');
colormap(jet);
h = colorbar;
ylabel(h,'E Field Magnitude [V/m]');
title(sprintf('Y Magnitude at Input of Aperture of Length =
%0.2f mm',L));

figure(5)
imagesc(y,z,PhaseZMatrix);
xlabel('Meters');
ylabel('Meters');
colormap(jet);
```

```
h = colorbar;
ylabel(h,'Phase in Degrees');
title(sprintf('Z Phase at Input of Aperture of Length =
%0.2f mm',L));

figure(6)
imagesc(y,z,MagZMatrix,[0,26]);
xlabel('Meters');
ylabel('Meters');
colormap(jet);
h = colorbar;
ylabel(h,'E Field Magnitude [V/m]'); %E Field [V/m]
title(sprintf('Z Magnitude at Input of Aperture of Length =
%0.2f mm',L));
```

A.3 Calculating the Hologram

A.3.1 Parent Script

```
%Emily Backer
%Calculation of Hologram

%% clear slate
close all;          % close figure windows
clear variables;    % clear memory
clc;                % clear command window
fclose('all');      % close files


%% Constants
c = physconst('lightspeed'); %299792458

%% Variables
f = 5.8e9; %5.8GHz
F = 10; %Distance to focal points, meters
L = 1.216; %Aperture Size LxL, meters
lambda = c/f;
FresnelRegionStart = 0.62*sqrt(L.^3 /lambda); %Lower Bound
for F
FresnelRegionEnd = 2*(L^2)/lambda; %Boundary for F
periodicity = 1.9e-2; %each patch periodicity of 1.9cm, 64
elements in a row -> 4,096 patches


%% Get Desired Aperture Pattern Ea
```

```
[Ea, UnwrappedEa,UnwrappedEaDeg] =
DesiredAperture(f,lambda,F,L,periodicity);
%Function calculates desired aperture pattern and plots
%Inputs: frequency, lambda, Focal Distance, Length of
Aperture, periodicity
%of elements
%Outputs: Ea = wrapped to 180 matrix, UnwrappedEa =
unwrapped Ea radians,
%UnwrappedEaDeg = unwrapped Ea degrees
MinEa = min(Ea);
MaxEa = max(Ea);
%% Get Horn Reference Wave
EVectorFileName = 'HornReferenceWaveV3-1.fld'; %filename
that has the extracted Efield Vector data
[Er,MagMatrix,Ephase,Emag,y,z] =
HornReferencePhase(L,periodicity,EVectorFileName);
%Function calculates the Ez magnitude and phase, puts it
into a matrix, and
%plots
%Inputs: Length of Aperture, periodicity of elements,
filename that
%contains the Efield Vector data
%Outputs: Er = matrix of phase values, MagMatrix = matrix
of mag values,
%Ephase = column of phase values, Emag = column of mag
values
%y: y coordinates
%z: z coordinates

%% Find Hologram
% H = Ea * Er^-1;
% Hphase = AperturePhase - ReferencePhase;
H = wrapTo180(Ea-Er);

figure
imagesc(y,z,H);
xlabel('Meters');
ylabel('Meters');
colormap(hsv);
h = colorbar;
ylabel(h,'Phase in Degrees'); %E Field [V/m]
title(sprintf('H Phase w/ L = %0.3f Meters',L));
set(gca,'YDir','normal')

%% Limit Hologram
% Limit hologram to the min and max achievable shift in
degrees
```

```
PatchCurvefilename =
'CirclePatchReflectionManyPointsMorePasses1.2-18.99.csv';
T = readtable(PatchCurvefilename);
Tdata = T.Variables;
SizeCurve = double(Tdata(:,1));
ReflPhaseCurve = double(Tdata(:,3));
ReflMagCurve = double(Tdata(:,2));

figure
imagesc(y,z,H,[min(ReflPhaseCurve) max(ReflPhaseCurve)]);
xlabel('Meters');
ylabel('Meters');
colormap(hsv);
h = colorbar;
ylabel(h,'Phase in Degrees');
title(sprintf('H Phase limited w/ L = %0.3f Meters',L));
set(gca,'YDir','normal') %Make the y axis go from - to +

%% Calculate Necessary Patch Size at each location on the
aperture
[H_PatchSizes,H_PatchSizesLineBased] =
CalculateHolPatchSizes(H,L,PatchCurvefilename,y,z);

%% Make HFSS Script
ProjectFileName = 'HFSS_Script_FR4'; %HFSS Project filename
ScriptFilename = 'HFSS_ApertureScript_FR4.py'; %Script
filename
% function assumes that the hfss design is title
HFSSDesign1
HFSS_Hologram_Script(ScriptFilename,
H_PatchSizes,ProjectFileName);
```

A.3.2 Functions Listed in Parent Script

A.3.2.1 Desired Aperture

```
%function to create desired aperture and plot

function [Ea,UnwrappedEa,UnwrappedEaDeg] =
DesiredAperture(f,lambda,F,L,periodicity)

%% Calculate Phase Right off of Aperture (Desired Aperture
Pattern Ea)
pixel = ceil(L/periodicity); %define how many
patches/pixels there will be
```

```
if mod(pixel,2)==1 %make pixel even number
    pixel = pixel-1;
end

Ea = zeros(pixel,pixel);
A = periodicity/2;
y = floor(-
pixel/2)*periodicity+A:periodicity:floor(pixel/2)*periodici
ty-A;
z = floor(pixel/2)*periodicity-A:-periodicity:floor(-
pixel/2)*periodicity+A;

for r = 1:pixel
    for c = 1:pixel
        Ea(r,c) = (2*pi/lambda)*(sqrt(z(r).^2 + y(c).^2 +
F^2)-F);
    end
end

figure
UnwrappedEa = Ea;
UnwrappedEaDeg = rad2deg(Ea);
Ea = wrapTo180(UnwrappedEaDeg);
imagesc(y,z,Ea);
xlabel('Meters');
ylabel('Meters');
colormap(hsv);
h = colorbar;
ylabel(h,'Phase in Degrees');
title(sprintf('Phase Needed at Output of Aperture for L =
%0.3f meters',L));
set(gca,'YDir','normal')
end
```

A.3.2.2 Calculate Horn Reference Phase

```
% Emily Backer
% import HFSS file and extract the phase information, plot,
and return Er
% file contains x from 0 to 19mm, y every 19mm, and z every
19mm
% x: start: 0        stop: 19mm     step: 19mm
% y: start: -598.5mm stop: 598.5mm   step: 19mm
% z: start: -598.5mm stop: 598.5mm   step: 19mm
```

```
function [PhaseMatrix,MagMatrix,EPhase,EMag,y,z] =
HornReferencePhase(L,Periodicity,EVectorFile)
% (1)import HFSS file and extract vector information
% (2)find phase information
% (3)plot

%% Aperture Info
%Periodicity = 19; %19mm, 64 elements in a row -> 4,096
patches
%L = 1216; %1.216meters, 1216mm
Pixel = Periodicity; %0.019meters, 19mm
numZpos = L/Pixel; %number of z and y positions,
numYpos = numZpos;

%% (1)Read in file and Extract
PatchInfo = dlmread(EVectorFile); %file contains complex E
vector data

%Extract x y z coordinates and corresponding phase
FirstHalf = 1:length(PatchInfo)/2';
x = PatchInfo(FirstHalf,1);
y = PatchInfo(FirstHalf,2);
z = PatchInfo(FirstHalf,3); %had to import x info at 0mm
and 1mm, but only need phase at 0mm
ExReal = PatchInfo(FirstHalf,4);
ExImag = PatchInfo(FirstHalf,5);
EyReal = PatchInfo(FirstHalf,6);
EyImag = PatchInfo(FirstHalf,7);
EzReal = PatchInfo(FirstHalf,8);
EzImag = PatchInfo(FirstHalf,9);

%% (2)Calculate Phase and Magnitude
EPhase = atan2d(EzImag,EzReal);
EMag = sqrt(EzReal.^2+EzImag.^2);

PatchPhaseMagnitude = EPhase;

%% Re-arrange into a Matrix for imagesc
% Z and Y increases, but Y increases slower
% i.e. if it is 40mm by 40mm array, -20mm to 20mm, Y
increases slower
% Y = -20mm, while Z -20mm:1mm:20mm, then Y = -19mm and Z
increases
% again -20mm:1mm:20mm
PhaseMatrix = zeros(numYpos,numZpos);
MagMatrix = zeros(numYpos,numZpos);
row = numYpos:-1:1;
```

```
col = 1:1:numZpos;
indexStart = 1;
indexStop = numZpos;
for c = col
    PhaseMatrix(row,c) =
PatchPhaseMagnitude(indexStart:indexStop);
    indexStart = indexStart + numZpos;
    indexStop = indexStop + numZpos;
end
indexStart = 1;
indexStop = numZpos;
for c = col
    MagMatrix(row,c) = EMag(indexStart:indexStop);
    indexStart = indexStart + numZpos;
    indexStop = indexStop + numZpos;
end
Er = PhaseMatrix;

%% (3)Plot
figure
imagesc(y,z,PhaseMatrix);
xlabel('Meters');
ylabel('Meters');
colormap(hsv);
h = colorbar;
ylabel(h,'Phase in Degrees'); %E Field [V/m]
title(sprintf('Z Phase w/ Length = %0.3f Meters',L)); %E
field at...
set(gca,'YDir','normal')

figure
imagesc(y,z,MagMatrix);
xlabel('Meters');
ylabel('Meters');
colormap(hsv);
h = colorbar;
ylabel(h,'E Field Magnitude [V/m]'); %E Field [V/m]
title(sprintf('Ez Abs w/ Length = %0.3f Meters',L)); %E
field at...
set(gca,'YDir','normal')
end
```

A.3.2.3 Calculate Hologram Patch Sizes

```
%Emily Backer
%Thesis - replace hologram phase with patch size
```

```
function [H_PatchSizes,H_PatchSizes_LineBased] =
CalculateHolPatchSizes(H,L,filename,y,z)

[row,col] = size(H);
H_PatchSizes_PointBased = zeros(size(H));
H_PatchSizes_LineBased = zeros(size(H));

%% Import sheet and Read Data
T = readtable(filename);
Tdata = T.Variables;
SizeCurve = double(Tdata(:,1));
ReflPhaseCurve = double(Tdata(:,3));
ReflMagCurve = double(Tdata(:,2));
PhaseDifference_PointBased = zeros(size(H));
%PhaseDifference_LineBased = zeros(size(H));

%% Graph Reflection Response Vs. Size
figure
subplot(2,1,1)
plot(SizeCurve,ReflMagCurve)
title('Magnitude of Reflection vs. Patch Size');
ylabel('Magnitude');
subplot(2,1,2)
plot(SizeCurve,ReflPhaseCurve)
title('Phase of Reflection vs. Patch Size');
xlabel('Patch Size [mm]');
ylabel('Phase [degrees]');

%% Find Patch Size Correlating to Phase
for r = 1:row
    for c = 1:col
        N = H(r,c);
        %Point Based
        [minDistance, indexOfMin] = min(abs(ReflPhaseCurve-
N));
        H_PatchSizes_PointBased(r,c) =
SizeCurve(indexOfMin);
        PhaseDifference_PointBased(r,c) = minDistance;

        %Line Based
        [distances, idx] = sort(abs(ReflPhaseCurve-N));
        idx = idx(1:2);
        Phase1 = ReflPhaseCurve(idx(1));
        Phase2 = ReflPhaseCurve(idx(2));
        Size1 = SizeCurve(idx(1));
        Size2 = SizeCurve(idx(2));
        m = (Phase1-Phase2)/(Size1-Size2);
```

```
        b = Phase1-(Size1)*m;
        PatchSize = (N-b)/m;
        H_PatchSizes_LineBased(r,c) = PatchSize;
    end
end

%% Graphs Point Based
figure
imagesc(y,z,H_PatchSizes_PointBased,[min(SizeCurve),max(Siz
eCurve)])
title(sprintf('Patch Size Needed w/ L = %0.3f meters',L));
h = colorbar;
ylabel(h,'Patch Size [mm]');
set(gca,'YDir','normal')

figure
imagesc(y,z,PhaseDifference_PointBased)
title('Error with Patch Size Phase and Phase Needed');
i = colorbar;
ylabel(i,'Error with Patch Size Phase [degrees]')
set(gca,'YDir','normal')

H_PatchSizes = H_PatchSizes_PointBased;
%% Graphs Line Based
figure
imagesc(y,z,H_PatchSizes_LineBased,[min(SizeCurve),max(Size
Curve)])
title(sprintf('Line Based Patch Size w/ L = %0.3f
meters',L));
h = colorbar;
ylabel(h,'Patch Size [mm]');
set(gca,'YDir','normal')

end
```

A.3.2.4 HFSS Script Parent function

```
% Hologram HFSS Script, creates in python
% Note: all numbers not in terms of variables must be in mm

function [filename] =
HFSS_Hologram_Script(filename,Lpatch,ProjectFileName)
%% Variables
L = 1.216; %1.216 meters
periodicity = 1.9e-2; %1.9cm
periodOv2 = periodicity/2;
```

```
pixel =L/periodicity; %64 rows and columns

%% Make File
fid = fopen(filename,'wt');

%% Intro texts
fprintf(fid, '#-----------------------------------------
----------\n');
fprintf(fid, '# Script Made by Emily Backer\n');
fprintf(fid, '#-----------------------------------------
----------\n');
fprintf(fid, 'import ScriptEnv\n');
fprintf(fid,
'ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")\n');
fprintf(fid, 'oDesktop.RestoreWindow()\n');
fprintf(fid, 'oProject =
oDesktop.SetActiveProject("%s")\n',ProjectFileName);
fprintf(fid, 'oDesign =
oProject.SetActiveDesign("HFSSDesign1")\n');
fprintf(fid, 'oEditor = oDesign.SetActiveEditor("3D
Modeler")\n');

%% Make Local Variables
%substrateH
substrateH_name = 'substrateH';
substrateH = 60;
units = 'mil';
createLocalVariable(fid,substrateH_name,substrateH,units);

%copperH
copperH_name = 'copperH';
copperH = 1.34;
units = 'mil';
createLocalVariable(fid,copperH_name,copperH,units);

%Laperture
Laperture_name = 'Laperture';
Laperture = 1.216;
units = 'meter';
createLocalVariable(fid,Laperture_name,Laperture,units);

%% Substrate
% L = 1.216 meters, FR4, SubstrateH = 60mils
% CornerPosition: 0, -L/2, -L/2
% x: -substrateH, y: L, z: L

Substrate_name = 'Substrate';
```

```
material = 'FR4_epoxy';
Xposition = '0mm';
Yposition = '-Laperture/2';
Zposition = '-Laperture/2';
Xdist = '-substrateH';
Ydist = 'Laperture';
Zdist = 'Laperture';
solveInside = 'True';

createBox(fid,Substrate_name,material,Xposition, Yposition,
Zposition, Xdist, Ydist, Zdist,solveInside);

%% GND_plane
% copperH = 1oz = 1.34mils, Copper
% CornerPosition: -substrateH,-L/2, -L/2
% x: -copperH, y: L, z: L

GND_plane_name = 'GND_plane';
material = 'copper';
Xposition = '-substrateH';
Yposition = '-Laperture/2';
Zposition = '-Laperture/2';
Xdist = '-copperH';
Zdist = 'Laperture';
Zdist = 'Laperture';
solveInside = 'False';

createBox(fid,GND_plane_name,material,Xposition, Yposition,
Zposition, Xdist, Ydist, Zdist,solveInside);

%% Make 4096 Patches
% Lpatch varies, copper, name: Patch(row)(col)
% Origin: 0, y(col)-Lpatch/2, z(row)-Lpatch/2
% x: copperH, y: Lpatch, z: Lpatch
y = -(pixel/2)*periodicity +
periodOv2:periodicity:(pixel/2)*periodicity - periodOv2;
z = (pixel/2)*periodicity - periodOv2:-periodicity:-
(pixel/2)*periodicity + periodOv2;
row = 1:1:pixel;
col = 1:1:pixel;

%Convert y z into mm;
y = y./(1e-3); %y in mm
z = z./(1e-3); %z in mm
%need to make sure the xorigin, yorigin, zorigin, and
radius are in the
%same units
```

```
for r = row
    for c = col
        LpatchRC = Lpatch(r,c);
        name = sprintf('Patch%d_%d',r,c);
        material = 'copper';
        Xorigin = 0;
        Yorigin = y(c);
        Zorigin = z(r);
        Axis = 'x';
        Radius = LpatchRC/2;
        units = 'mm';
        Height = 'copperH';

createCylinder(fid,name,material,Xorigin,Yorigin,Zorigin,Ax
is,Radius,units,Height,solveInside);

    end
end

fclose(fid);
end
```

createLocalVariable Function

```
function createLocalVariable(fid,name,value,units)
fprintf(fid, 'oDesign.ChangeProperty(\n\t[\n');
fprintf(fid, '\t\t"NAME:AllTabs",\n\t\t[\n');
fprintf(fid, '\t\t\t"NAME:LocalVariableTab",\n\t\t\t[\n');
fprintf(fid, '\t\t\t\t"NAME:PropServers", \n');
fprintf(fid,
'\t\t\t\t"LocalVariables"\n\t\t\t],\n\t\t\t[\n');
fprintf(fid, '\t\t\t\t"NAME:NewProps",\n\t\t\t\t[\n');
fprintf(fid, '\t\t\t\t\t"NAME:%s",\n',name);
fprintf(fid, '\t\t\t\t\t"PropType:="\t\t,
"VariableProp",\n');
fprintf(fid, '\t\t\t\t\t"UserDef:=" \t, True,\n');
fprintf(fid, '\t\t\t\t\t"Value:="\t\t,
"%f%s"\n',value,units);
fprintf(fid, '\t\t\t\t]\n\t\t\t]\n\t\t]\n\t])\n');
end
```

createBox Function

```
% Function that creates box in script
function createBox(fid, name, material, Xposition,
Yposition, Zposition, Xdist, Ydist, Zdist,solveInside)
fprintf(fid, 'oEditor.CreateBox(\n\t[\n');
```

```
fprintf(fid, '\t\t"NAME:BoxParameters",\n');
fprintf(fid, '\t\t"XPosition:="\t\t, "%s",\n',Xposition);
fprintf(fid, '\t\t"YPosition:="\t\t, "%s",\n',Yposition);
fprintf(fid, '\t\t"ZPosition:="\t\t, "%s",\n',Zposition);
fprintf(fid, '\t\t"XSize:="\t\t, "%s",\n',Xdist);
fprintf(fid, '\t\t"YSize:="\t\t, "%s",\n',Ydist);
fprintf(fid, '\t\t"ZSize:="     , "%s"\n',Zdist);
fprintf(fid, '\t],\n\t[\n');
fprintf(fid, '\t\t"NAME:Attributes",\n');
fprintf(fid, '\t\t"Name:=" \t\t, "%s",\n',name);
fprintf(fid, '\t\t"Flags:="\t\t, "",\n');
fprintf(fid, '\t\t"Color:="\t\t, "(143 175 143)",\n');
fprintf(fid, '\t\t"Transparency:="\t, 0,\n');
fprintf(fid, '\t\t"PartCoordinateSystem:=", "Global",\n');
fprintf(fid, '\t\t"UDMId:="\t\t, "",\n');
fprintf(fid, '\t\t"MaterialValue:="\t,
"\\"%s\\"",\n',material);
fprintf(fid, '\t\t"SurfaceMaterialValue:=", "\\"\\"",\n');
fprintf(fid, '\t\t"SolveInside:="\t\t, %s,\n',solveInside);
fprintf(fid, '\t\t"IsMaterialEditable:="\t, True,\n');
fprintf(fid, '\t\t"UseMaterialAppearance:=", False,\n');
fprintf(fid, '\t\t"IsLightweight:="\t, False\n');
fprintf(fid, '\t])\n');
end
```

A.3.2.4.3 createCylinder Function

```
function
createCylinder(fid,name,material,Xorigin,Yorigin,Zorigin,Ax
is,Radius,units,Height,solveInside)
fprintf(fid, 'oEditor.CreateCylinder(\n');
fprintf(fid, '\t[\n');
fprintf(fid, '\t\t"NAME:CylinderParameters",\n');
fprintf(fid, '\t\t"XCenter:="\t\t,
"%f%s",\n',Xorigin,units);
fprintf(fid, '\t\t"YCenter:="\t\t,
"%f%s",\n',Yorigin,units);
fprintf(fid, '\t\t"ZCenter:="\t\t,
"%f%s",\n',Zorigin,units);
fprintf(fid, '\t\t"Radius:="\t\t, "%f%s",\n',Radius,units);
fprintf(fid, '\t\t"Height:="\t\t, "%s",\n',Height);
fprintf(fid, '\t\t"WhichAxis:=" \t, "%s",\n',Axis);
fprintf(fid, '\t\t"NumSides:="  \t, "0"\n');
fprintf(fid, '\t],\n\t[\n');
fprintf(fid, '\t\t"NAME:Attributes",\n');
fprintf(fid, '\t\t"Name:="      , "%s",\n',name);
fprintf(fid, '\t\t"Flags:="     , "",\n');
```

```
fprintf(fid, '\t\t"Color:="     , "(255 128 0)",\n');
fprintf(fid, '\t\t"Transparency:="  , 0,\n');
fprintf(fid, '\t\t"PartCoordinateSystem:=", "Global",\n');
fprintf(fid, '\t\t"UDMId:="     , "",\n');
fprintf(fid, '\t\t"MaterialValue:=" ,
"\\"%s\\"",\n',material);
fprintf(fid, '\t\t"SurfaceMaterialValue:=", "\\"\\"",\n');
fprintf(fid, '\t\t"SolveInside:="       ,
%s,\n',solveInside);
fprintf(fid, '\t\t"IsMaterialEditable:="    , True,\n');
fprintf(fid, '\t\t"UseMaterialAppearance:=", False,\n');
fprintf(fid, '\t\t"IsLightweight:=" , False\n');
fprintf(fid, '\t])\n');
end
```

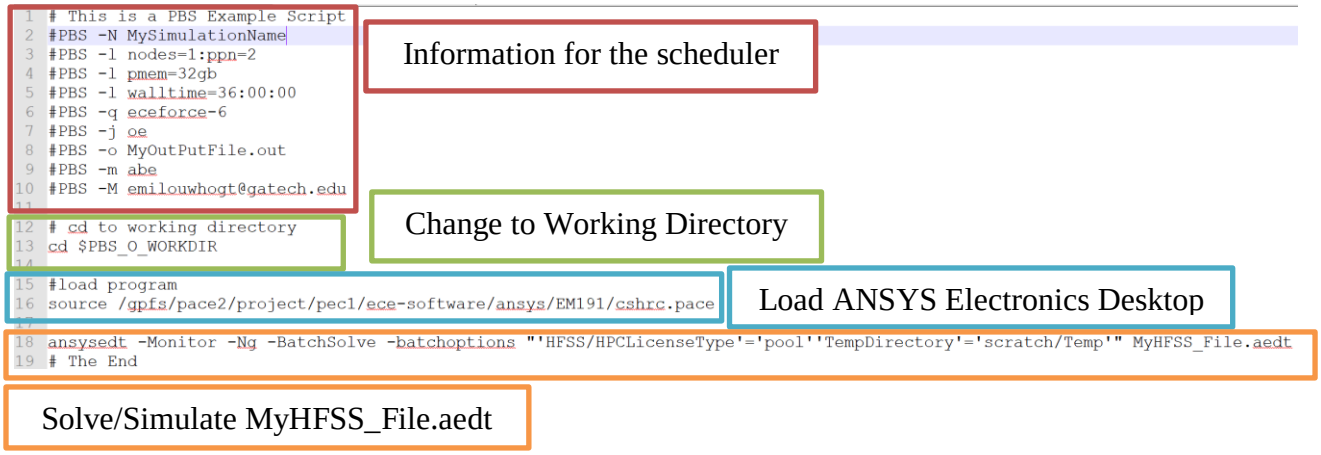
Useful link: https://arrc.ou.edu/~cody/hfsslib/intro/

APPENDIX B. USING THE PACE

The Partnership for an Advanced Computing Environment (PACE) provides faculty, researchers, and students a cluster of computers. The PACE clusters include a mix of nodes from different departments, so a queue can be a small part of a cluster or span multiple clusters [43]. To access the clusters, first students will need to get an account, which can be done with one's advisor and the ECE help desk. After the user has an account, an SSH client such as PuTTY (for Windows) is needed. If the user wants to access the GUI/Interactive session of PACE, Xming or X-win32 is needed. In addition, when accessing PACE off the Georgia Tech network, a VPN is needed. WinSCP is also a useful application that transfers files between the PACE account and the user's computer. The following steps shows the user how to make a script for a batch job (submit and forget).

1. You will need a text editor. I use Notepad ++
2. Please see the following example script 1 for HFSS in the ANSYS Electronics Desktop Suite 19.1 Version. It is confusing because the electronics desktop will say 2018.1, but when you are downloading/accessing it from the cluster, the suite is 19.1. HFSS 19.1 module does not exist yet on the ecemodules, so you must use the ANSYS EM191 module for the ANSYS EM Suite 19.1 to access HFSS 2018.1

```
# This is a PBS Example Script
#PBS -N MySimulationName
#PBS -l nodes=1:ppn=2
#PBS -l pmem=32gb
#PBS -l walltime=36:00:00
#PBS -q eceforce-6
#PBS -j oe
#PBS -o MyOutPutFile.out
#PBS -m abe
#PBS -M emilouwhogt@gatech.edu

# cd to working directory
cd $PBS_O_WORKDIR

#load program
source /gpfs/pace2/project/pec1/ece-software/ansys/EM191/cshrc.pace

ansysedt -Monitor -Ng -BatchSolve -batchoptions "'HFSS/HPCLicenseType'='pool''TempDirectory'='scratch/Temp'" MyHFSS_File.aedt
# The End
```

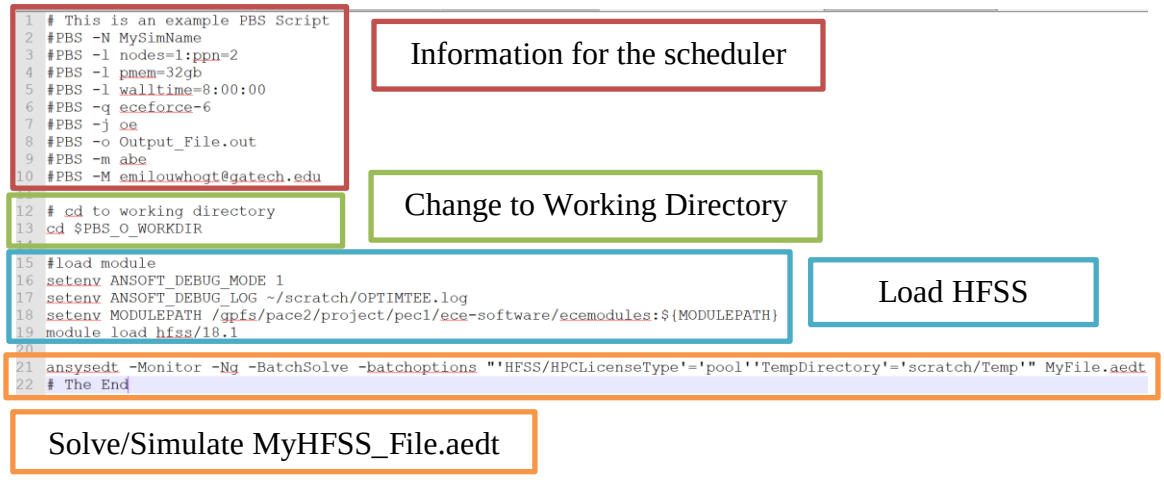



```
# This is a PBS Example Script
#PBS -N MySimulationName
#PBS -l nodes=1:ppn=2
#PBS -l pmem=32gb
#PBS -l walltime=36:00:00
#PBS -q eceforce-6
#PBS -j oe
#PBS -o MyOutPutFile.out
#PBS -m abe
#PBS -M emilouwhogt@gatech.edu

# cd to working directory
cd $PBS_O_WORKDIR

#load program
source /gpfs/pace2/project/pec1/ece-software/ansys/EM191/cshrc.pace

ansysedt -Monitor -Ng -BatchSolve -batchoptions "'HFSS/HPCLicenseType'='pool''TempDirectory'='scratch/Temp'" MyHFSS_File.aedt
# The End
```

1. Comment
2. Name of the batch/run
3. Requesting 1 node, 2 cores per node
4. Requesting 32GB memory per core (total of 64GB)
5. How much time my simulation needs (max time)
    a. The sim will stop running when it hits this walltime, make sure you provide enough time
6. Which queue I want to submit to
7. Put the output error file in specified format
8. Output error will go into a file named MyOutPutFile.out
9. Notify me on start finish and error
10. Notify me via this email
11. space, the following lines after 10 are where the actual computation happens
12. comment: change to working directory
13. change to working directory
14. space

15. comment: load the program (in this case ANSYS Electronics Desktop 19.1)
16. Load program (this might be different for different versions of ANSYS)
17. Space
18. Open ansysedt, monitor it, but no graphics, solve all, myHFSS_Filename.aedt
19. End of example script. According to another tutorial, it is necessary. Please see the below link to another tutorial
    a. https://drive.google.com/drive/folders/0B1D6uQMBGYO4a25yS0tiN1NJcHM?usp=sharing

3. To use HFSS older version, you can load the HFSS module
4. Please see the example script 2 below




```
1  # This is an example PBS Script
2  #PBS -N MySimName
3  #PBS -l nodes=1:ppn=2
4  #PBS -l pmem=32gb
5  #PBS -l walltime=8:00:00
6  #PBS -q eceforce-6
7  #PBS -j oe
8  #PBS -o Output_File.out
9  #PBS -m abe
10 #PBS -M emilouwhogt@gatech.edu
11
12 # cd to working directory
13 cd $PBS_O_WORKDIR
14
15 #load module
16 setenv ANSOFT_DEBUG_MODE 1
17 setenv ANSOFT_DEBUG_LOG ~/scratch/OPTIMTEE.log
18 setenv MODULEPATH /gpfs/pace2/project/pec1/ece-software/ecemodules:${MODULEPATH}
19 module load hfss/18.1
20
21 ansysedt -Monitor -Ng -BatchSolve -batchoptions "'HFSS/HPCLicenseType'='pool''TempDirectory'='scratch/Temp'" MyFile.aedt
22 # The End
```

i. In the above script, lines 1-10 are similar to the first example script. These lines tell the scheduler what memory and time requirements

are needed. It also can tell the scheduler to send an email on start and finish/error

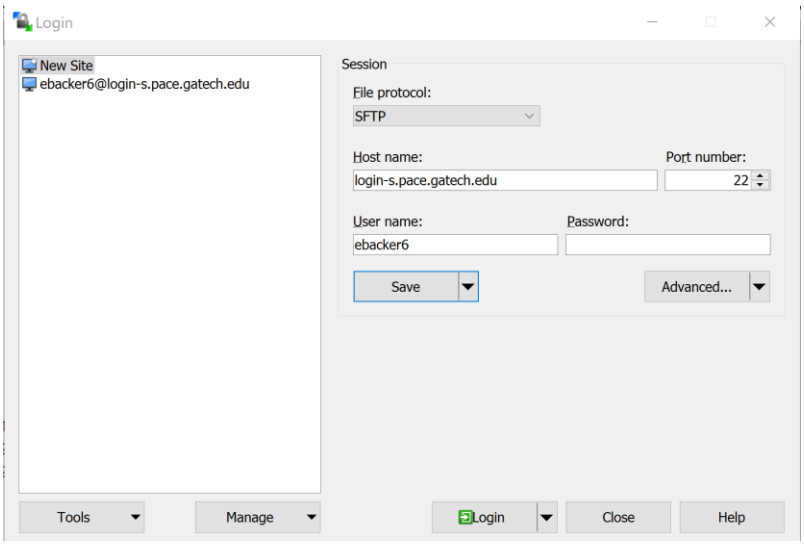
ii. Line 12-13: set working directory

iii. Line 15-19: load the module

iv. Line 21: open ansysedt, monitor the job, no graphics, solve all, temporary scratch directory, and finally the name of the ANSYS EDT file/HFSS file.

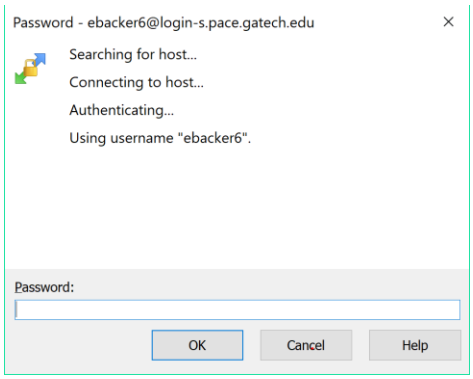
Now that a script has been made, the file needs to transfer from your computer/desktop to your PACE file folder. You have three storage directories: home, data and scratch. I store all my files in data since it is backed up daily. If you need to store more than 2 million files (or 100G depending on your department), you might need to use the scratch drive, which has a 7TB quota. The scratch drive is not backed up though, so I would only temporarily store files in the scratch drive. You can easily check your quota by typing pace-quota once you have logged into PutTTY, which I will go through later.

1. First make sure you have WinSCP installed
2. Open WinSCP and type in:
    a. Host name: <headnode>.pace.gatech.edu
    b. User name: <GaTech_ID>
    c. Password: your password

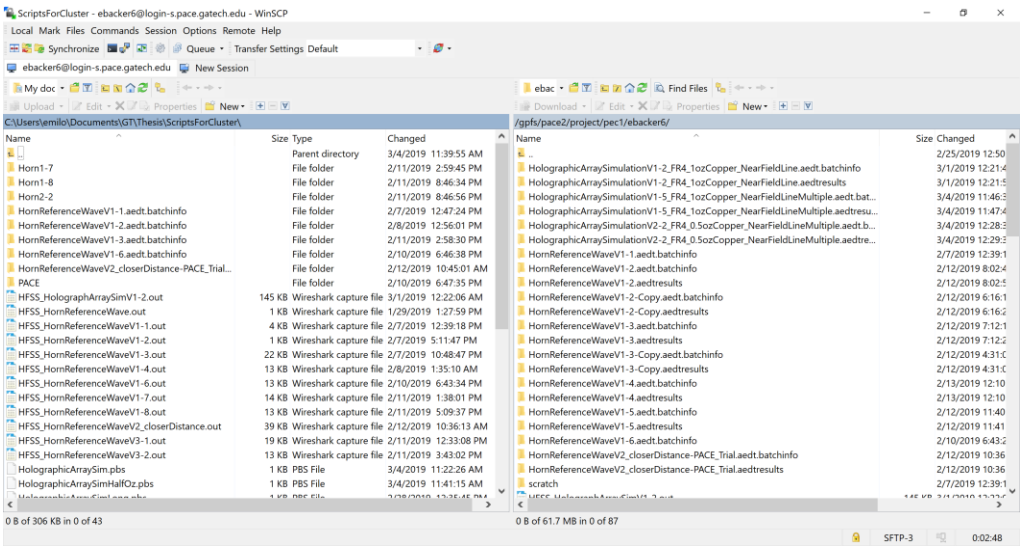
d. 

e. You can save this session, so next time, you only need to type in your password. Click Login

f. After you save the session, the next time you open WinSCP, after clicking login a window will pop up for you to type your password in as shown below.

i. 

3. Once you have successfully logged in, your window should look similar to the following:

a. 

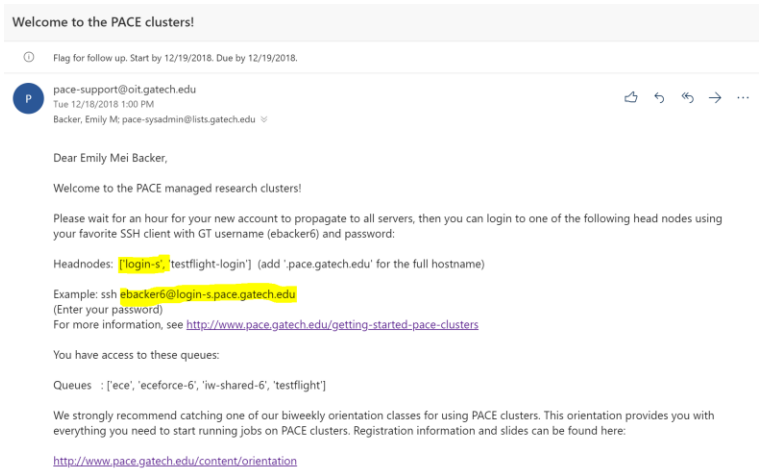
4. In each separate side, you can change the folders that you want to move files around from/to and organize the file directories.
5. Once you have the correct file folders, simply click and drag files from one side to the other

The following steps walk the user through connecting to PACE through PuTTY and submitting a batch job.

1. First decide if you wish to access the GUI or just submit a batch job. If you want to just submit a batch job, skip step 2, step 5, and step 8
2. Open Xming.
    a. Make sure there is an icon that indicates xming is running (might be in the hidden icons)
3. Open PuTTY.
4. After opening PuTTY, type in one's login information.
    a. <GT_user_ID>@<headnode>.pace.gatech.edu

i. The headnode will be in the email that the PACE department sends you that your account has been set up

ii.

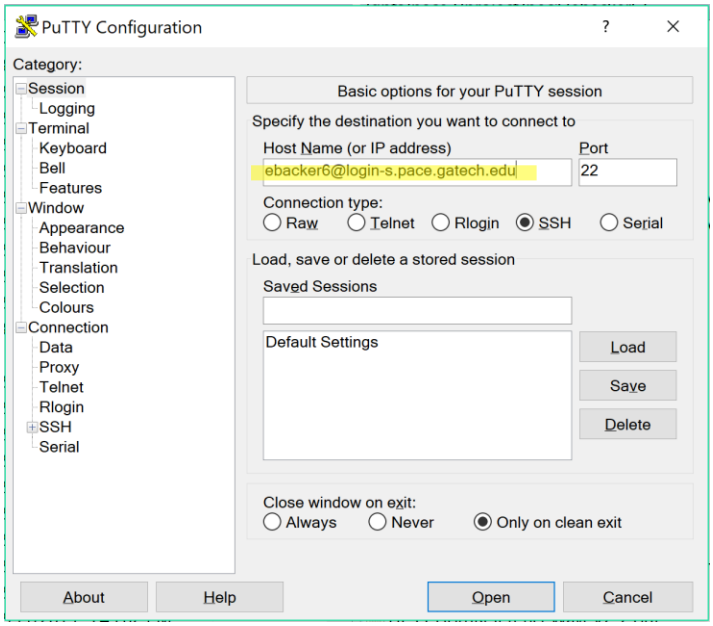


1. Email from PACE OIT

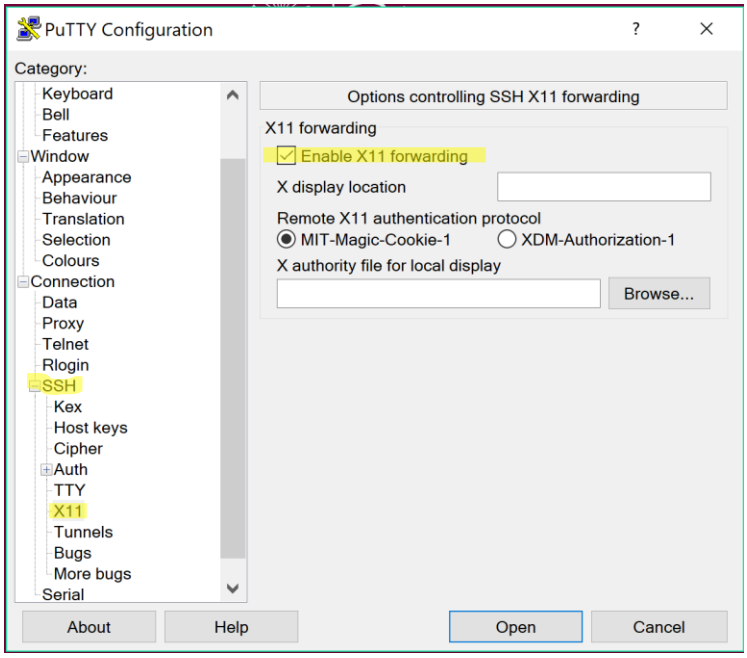
iii. In the above screen capture

1. Gt username: ebacker6
2. Headnode: login-s

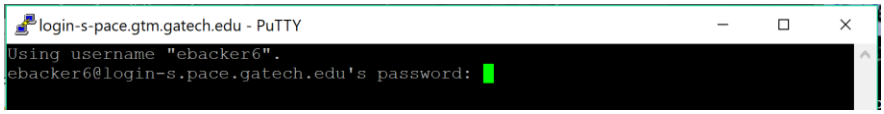
b.



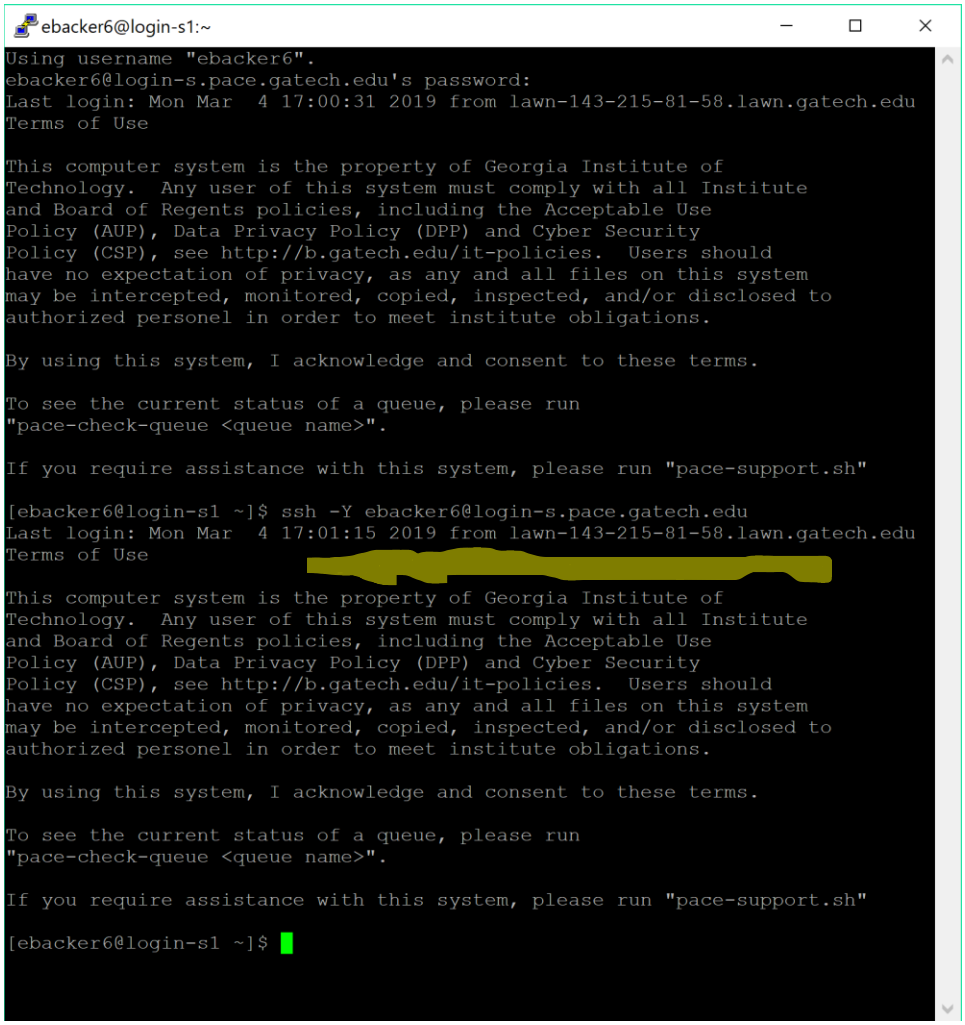
5. If one wishes to access the GUI, click the +SSH, X11, then check Enable X11 forwarding

a. 

6. Now click open
7. Type in your Georgia Tech password

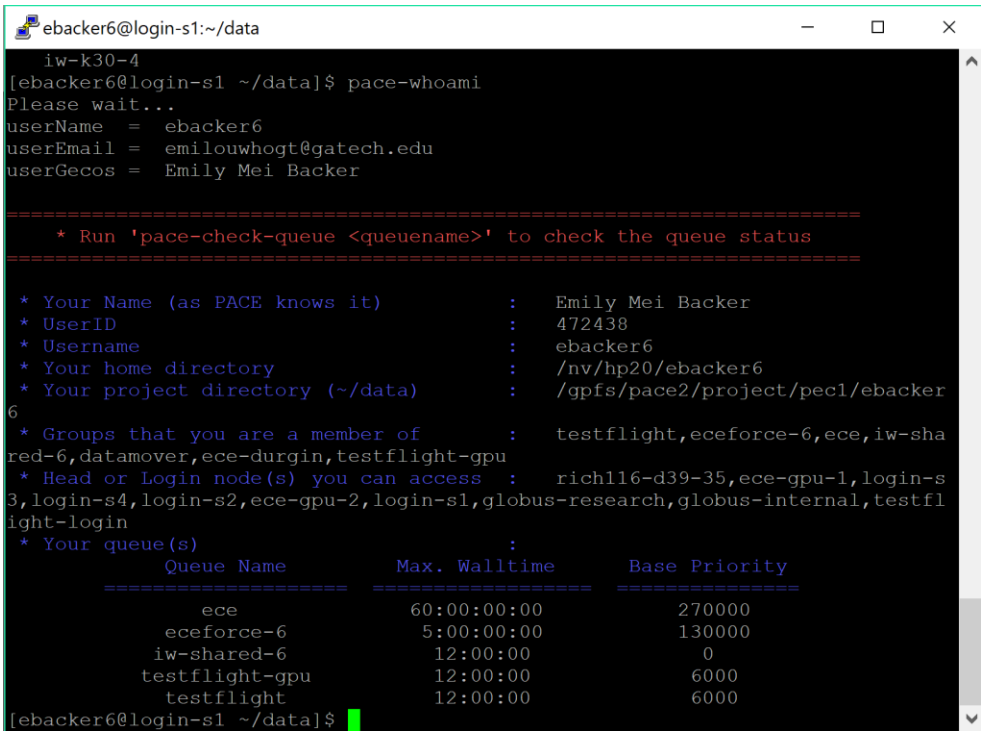
a. 

8. To connect to the GUI type the following
    a. ssh -Y <GT_user_ID>@<headnode>.pace.gatech.edu
    b. i.e. ssh -Y ebacker6@login-s.pace.gatech.edu

c. 

9. First time logging in, you might want to check which queues you can submit to and verify the information on your account. You can type:
    a. pace-whoami

b.



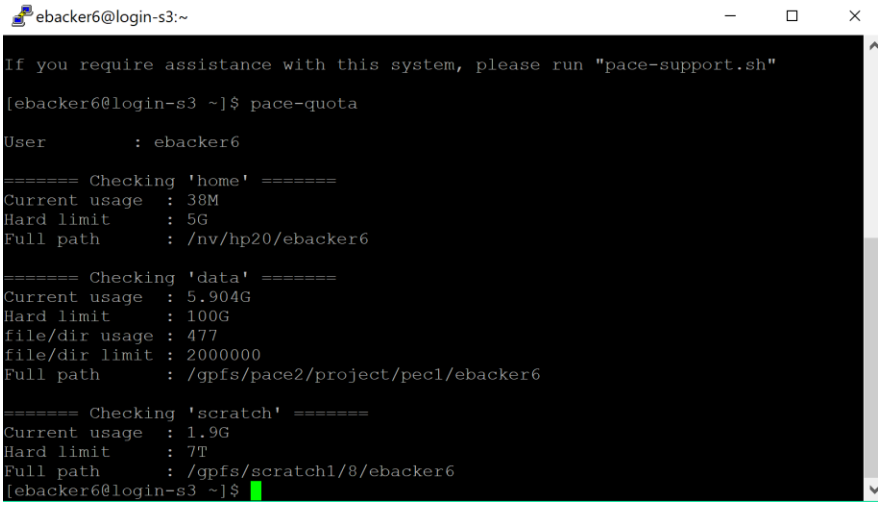
c. For quick jobs with high memory, iw-shared-6 is great to submit to, and it has a wall time of 12 hours.

d. Longer jobs (5 day max), eceforce-6 is the next best.

10. As mentioned before, you can check your storage by typing:

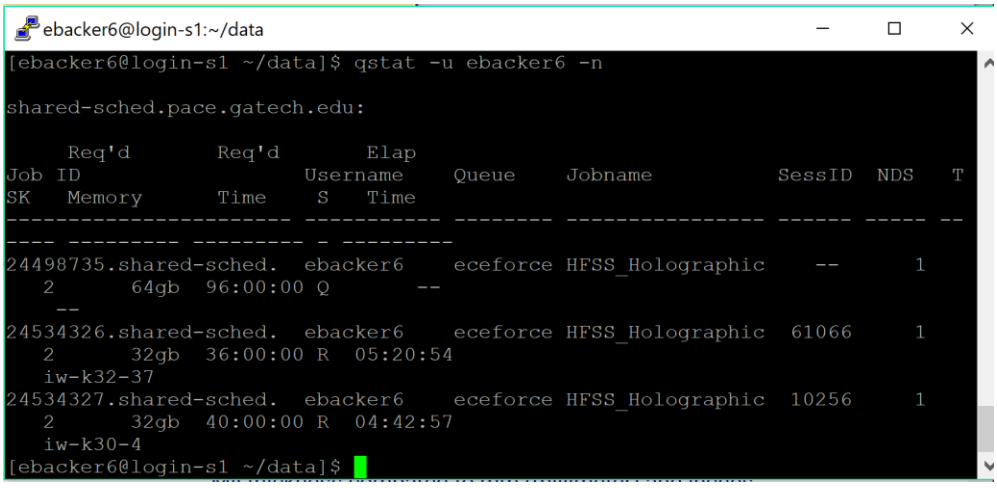
a. pace-quota

b.



11. To submit a job via script

a. Go to the directory your script is located. Most likely it will be in your data folder. Type:
    i. cd <your folder your script is located in>
    ii. i.e. cd data
b. If you want to look at the files in your data folder, type:
    i. ls
c. If you are using a windows computer, usually you will have to convert the file, type:
    i. dos2unix myscript.pbs
d. Now, submit the script, type:
    i. qsub myscript.pbs
e. once the script has been submitted, a job number should be returned

12. to check on your queue, you can type:
    a. qstat -u <GT_username> -n
    b. 

    c. I have three jobs currently submitted. Two are running, and one is queued.

13. To check your submitted jobs you can also type:
    a. showq | grep <GaTech_ID>

b.

```
[ebacker6@login-s1 ~/data]$ showq | grep ebacker6
24534326          ebacker6    Running     2  1:06:21:22  Mon Mar  4 11:46:23
24534327          ebacker6    Running     2  1:10:59:19  Mon Mar  4 12:24:20
24498735          ebacker6       Idle     2  4:00:00:00  Thu Feb 28 12:37:15
[ebacker6@login-s1 ~/data]$
```

14. To check a job

    a. checkjob <jobID>

b.

ebacker6@login-s1:~/data

```
[ebacker6@login-s1 ~/data]$ checkjob 24498735
job 24498735

AName: HFSS_HolographicArraySim
State: Idle
Creds:  user:ebacker6  group:ece-durgin  class:eceforce-6
WallTime:   00:00:00 of 4:00:00:00
BecameEligible: Fri Mar  1 18:29:43
SubmitTime: Thu Feb 28 12:37:15
  (Time Queued  Total: 4:04:41:33  Eligible: 4:04:40:56)

TemplateSets:  DEFAULT
Total Requested Tasks: 2

Req[0]  TaskCount: 2  Partition: ALL
Memory >= 64G  Disk >= 0  Swap >= 0
Dedicated Resources Per Task: PROCS: 1  MEM: 64G


SystemID:   Moab
SystemJID:  24498735
Notification Events: JobStart,JobEnd,JobFail  Notification Address: emilouwhog
t@gatech.edu

BypassCount:     23904
Partition List: pbs
Flags:          RESTARTABLE
Attr:           checkpoint
StartPriority:  136040
PE:             22.10
rejected for Class        - (null)
rejected for CPU          - (null)
rejected for Memory       - (null)
rejected for State        - (null)
NOTE:  job req cannot run in partition pbs (available procs do not meet requir
ements : 0 of 2 procs found)
idle procs: 4894  feasible procs:   0

Node Rejection Summary: [Class: 212][CPU: 1][Memory: 121][State: 230]


[ebacker6@login-s1 ~/data]$
```

    i. My job is currently idle: it is awaiting resources to become available

    c. https://pace.gatech.edu/how-do-i-check-status-job

    i. This link tells more about what checkjob tells you

15. To delete a job

    a. qdel <jobID>

Useful links:

- Orientation: http://www.pace.gatech.edu/content/orientation
- Orientation Slides:

  http://www.pace.gatech.edu/sites/default/files/pace_orientation_3.pdf

REFERENCES

[1] K. Byron, S. A. Winkler, F. Robb, S. Vasanawala, J. Pauly, and G. Scott, "An MRI Compatible RF MEMs Controlled Wireless Power Transfer System," *IEEE Transactions on Microwave Theory and Techniques,* pp. 1-10, 2019.

[2] L. L. Pon, C. Y. Leow, S. K. A. Rahim, A. A. Eteng, and M. R. Kamarudin, "Printed Spiral Resonator for Displacement-tolerant Near-field Wireless Energy Transfer," *IEEE Access,* pp. 1-1, 2019.

[3] X. Rui-Feng, K. P. Basappa, C. Hyouk-Kyu, Z. Yuanjin, and J. Minkyu, "High-efficiency transcutaneous wireless energy transfer for biomedical applications," in *2011 Defense Science Research Conference and Expo (DSR)*, 2011, pp. 1-4.

[4] Z. Zhang and B. Zhang, "Angular-Misalignment Insensitive Omnidirectional Wireless Power Transfer," *IEEE Transactions on Industrial Electronics,* pp. 1-1, 2019.

[5] F. F. Mendonça, D. A. Andrade, A. O. Cassio, G. P. Viajante, R. T. Fidelis, and E. N. Chaves, "Analysis of the Performance in Wireless Transmission Systems for Application in Electric Vehicles," in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, 2018, pp. 426-432.

[6] J. O. McSpadden and J. C. Mankins, "Space solar power programs and microwave wireless power transmission technology," *IEEE Microwave Magazine,* vol. 3, no. 4, pp. 46-57, 2002.

[7] B. Ratni, S. N. Burokur, A. d. Lustrac, D. Guihard, and B. Rmili, "Directive Reconfigurable Fabry-Perot Cavity Antenna for Space Applications," in *2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2018, pp. 104-106.

[8] W. Xin, H. Xinbin, W. Li, and L. Mingyu, "Employing phase-conjugation antenna array to beam microwave power from satellite to earth," in *2015 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2015, pp. 1-5.

[9] S. Aldhaher, P. D. Mitcheson, J. M. Arteaga, G. Kkelis, and D. C. Yates, "Light-weight wireless power transfer for mid-air charging of drones," in *2017 11th European Conference on Antennas and Propagation (EUCAP)*, 2017, pp. 336-340.

[10] J. A. L. Calvo, G. Alirezaei, and R. Mathar, "Wireless powering of drone-based MANETs for disaster zones," in *2017 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2017, pp. 98-103.

[11] C. H. Choi, H. J. Jang, S. G. Lim, H. C. Lim, S. H. Cho, and I. Gaponov, "Automatic wireless drone charging station creating essential environment for continuous drone operation," in *2016 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2016, pp. 132-136.

[12] K. Sang-Won, C. In-Kui, and H. Sung-Yong, "Comparison of charging region differences according to receiver structure in drone wireless charging system," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, 2017, pp. 1058-1060.

[13] S. Tansuriyong, M. Kyan, K. Numata, S. Taira, and T. Anezaki, "Verification experiment for drone charging station using RTK-GPS," in *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2017, pp. 229-232.

[14] J. Zhou, B. Zhang, W. Xiao, D. Qiu, and Y. Chen, "Nonlinear Parity-Time-Symmetric Model for Constant Efficiency Wireless Power Transfer: Application to a Drone-in-Flight Wireless Charging Platform," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 4097-4107, 2019.

[15] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*. John Wiley & Sons, Inc., 2012, p. 848.

[16] C. A. Balanis, *Antenna Theory Analysis and Design*, Third ed. Wiley-Interscience, p. 1136.

[17] S. Hekal *et al.*, "Compact Wireless Power Transfer System Using Defected Ground Bandstop Filters," *IEEE Microwave and Wireless Components Letters*, vol. 26, no. 10, pp. 849-851, 2016.

[18] F. Jolani, Y. Yu, and Z. Chen, "A Planar Magnetically Coupled Resonant Wireless Power Transfer System Using Printed Spiral Coils," *IEEE Antennas and Wireless Propagation Letters*, vol. 13, pp. 1648-1651, 2014.

[19] A. P. Sample, D. T. Meyer, and J. R. Smith, "Analysis, Experimental Results, and Range Adaptation of Magnetically Coupled Resonators for Wireless Power Transfer," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 2, pp. 544-554, 2011.

[20] W. C. Brown, "The History of Power Transmission by Radio Waves," *IEEE Transactions on Microwave Theory and Techniques*, vol. 32, no. 9, pp. 1230-1242, 1984.

[21] V. R. Gowda, O. Yurduseven, G. Lipworth, T. Zupan, M. S. Reynolds, and D. R. Smith, "Wireless Power Transfer in the Radiative Near Field," *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1865-1868, 2016.

[22] A. Buffi, P. Nepa, and G. Manara, "Design Criteria for Near-Field-Focused Planar Arrays," *IEEE Antennas and Propagation Magazine,* vol. 54, no. 1, pp. 40-50, 2012.

[23] M. Bogosanovic and A. G. Williamson, "Microstrip Antenna Array With a Beam Focused in the Near-Field Zone for Application in Noncontact Microwave Industrial Inspection," *IEEE Transactions on Instrumentation and Measurement,* vol. 56, no. 6, pp. 2186-2195, 2007.

[24] S. H. Zainud-Deen, H. A. Malhat, and K. H. Awadalla, "8x8 Near-Field Focused Circularly Polarized Cylindrical DRA Array for RFID Applications," *Electrical and Electronic Engineering,* Electrical and Electronic Engineering vol. 1, no. 1, pp. 42-48, 2012.

[25] D. R. Smith *et al.*, *An analysis of beamed wireless power transfer in the Fresnel zone using a dynamic, metasurface aperture*. Journal of Applied Physics, 2017.

[26] J. T. Verdeyen, *Laser Electronics*, 3rd ed. Pearson, 1995.

[27] C. Hou-Tong, J. T. Antoinette, and Y. Nanfang, "A review of metasurfaces: physics and applications," *Reports on Progress in Physics,* vol. 79, no. 7, p. 076401, 2016.

[28] D. Gabor, "A New Microscopic Principle," *Nature,* vol. 161, p. 777, 05/15/online 1948.

[29] G. S. Lipworth *et al.*, "A Large Planar Holographic Refiectarray for Fresnel-Zone Microwave Wireless Power Transfer at 5.8 GHz," in *2018 IEEE/MTT-S International Microwave Symposium - IMS*, 2018, pp. 964-967.

[30] J. B. Pendry, A. J. Holden, D. J. Robbins, and W. J. Stewart, "Magnetism from conductors and enhanced nonlinear phenomena," *IEEE Transactions on Microwave Theory and Techniques,* vol. 47, no. 11, pp. 2075-2084, 1999.

[31] G. Lipworth, N. W. Caira, S. Larouche, and D. R. Smith, "Phase and magnitude constrained metasurface holography at W-band frequencies," *Optics Express,* vol. 24, no. 17, 2016.

[32] D. Berry, R. Malech, and W. Kennedy, "The reflectarray antenna," *IEEE Transactions on Antennas and Propagation,* vol. 11, no. 6, pp. 645-651, 1963.

[33] D. F. Sievenpiper, J. H. Schaffner, H. J. Song, R. Y. Loo, and G. Tangonan, "Two-dimensional beam steering using an electrically tunable impedance surface," *IEEE Transactions on Antennas and Propagation,* vol. 51, no. 10, pp. 2713-2722, 2003.

[34] "Varactor Diodes Application Note," datasheet p. 6, 2008.

[35] B. O. Zhu, J. Zhao, and Y. Feng, "Active impedance metasurface with full 360° reflection phase tuning," *Scientific Reports,* Article vol. 3, p. 3059, 10/28/online 2013.

[36] I. V. Shadrivov, S. K. Morrison, and Y. S. Kivshar, "Tunable split-ring resonators for nonlinear negative-index metamaterials," *Optics Express,* vol. 14, no. 20, pp. 9344-9349, 2006/10/02 2006.

[37] M. U. Afzal and K. P. Esselle, "High-Gain Beam Steering by Near-Field Phase Transformation - An Overview," in *2018 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting*, 2018, pp. 1447-1448.

[38] ANSYS, *Getting Started with HFSS: Floquet Ports*. Canonsburg, Pennsylvania: SAS IP, Inc., 2016, p. 64.

[39] P. E. Inc, "WR-159 Waveguide Standard Gain Horn Antenna Operating From 4.9GHz to 7.05 GHz With a Nominal 15dB Gain SMA Female Input " datasheet no. PE9860-15, 2018.

[40] D. Edgar, "HFSS 13: Hybrid FE-BI for Efficient Simulation of Radiation and Scattering," 2011, Ansys, Inc.

[41] J. Silvestro, "Hybrid Finite Element Boundary Integral Method," p. 9 White Paper, 2010, Ansys, Inc., Canonsburg, PA, USA

[42] C. Piersall. *Introduction HFSS Scripts*, 2015. [Online]. Available: https://arrc.ou.edu/~cody/hfsslib/intro/. [Accessed March 15, 2019].

[43] S. Sarajilic and M. Belgin, "PACE Orientation," OIT Georgia Institute of Technology, Atlanta, GA, 2019.